Statistics-based Rule Generation for Filipino Style and Grammar Checking

A Thesis
Presented to
the Faculty of the College of Computer Studies
De La Salle University

In Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Computer Science

by
Oco, Nathaniel A.

Joel P. Ilao, PhD
Faculty Adviser

Acknowledgment

I would like to acknowledge the following for being instrumental to the completion of this research:

- Thesis adviser – Dr. Joel Ilao – for guiding me in every step [of the way];
- Panel leader – Dr. Rachel Edita Roxas – who introduced me to the world of research and continuously provide opportunities for me to grow;
- Panel member – Mr. Allan Borra – for sharing his knowledge and time, and believing in person's capability to achieve higher heights;
- Friend – Mr. Leif Romeritch Syliongka – for all our idea-bouncing sessions (n.b., they generate novel and disruptive outputs) and for constantly reminding me that everything will turn out well;
- Friend – Mr. Jason Wong – whose programming skills can start a company, he will feed me and Leif one of these days; and
- Friends – Mr. Alron Jan Lam and Mr. Ivan Paner – who made me reaffirm my belief that there is "vacation" in research activities.

To all I failed to mention but should be in this list, thank you.

皆さん、
いろいろお世話になっています。

Partial results of this research have been published in the following:

Nathaniel Oco, Joel Ilao, Rachel Edita Roxas, Leif Romeritch Syliongka. 2013. Measuring Language Similarity using Trigrams: Limitations of Language Identification. In Proceedings of the 3rd International Conference on Recent Trends in Information Technology (Chennai, India. July 25-27, 2013).

Nathaniel Oco, Leif Romeritch Syliongka, Joel Ilao, Rachel Edita Roxas. 2013. Dice's Coefficient on Trigram Profiles as Metric for Language Similarity. In Proceedings of the 16th Oriental COCOSDA (Gurgaon, India. November 25-27, 2013). Published by IEEE Xplore. ISBN: 978-147992378-6. http://dx.doi.org/10.1109/ICSDA.2013.6709892

Nathaniel Oco, Leif Romeritch Syliongka, Joel Ilao, and Rachel Edita Roxas. 2014a. N-gram based Language Identification and Rule-based Grammar Checking. In Proceedings of the 14th Philippine Computing Science Congress (Davao City, Philippines, March 06-08, 2014). Published by the Computing Society of the Philippines, pp 244-250.

Nathaniel Oco, Raquel Sison-Buban, Leif Romeritch Syliongka, Rachel Edita Roxas, and Joel Ilao. 2014b. Ang Paggamit ng Trigram Ranking Bilang Panukat sa Pagkakahalintulad at Pagkakapangkat ng mga Wika/Trigram Ranking: Metric for Language Similarity and Clustering. Malay 26 (2), pp 53-68.

Abstract
Current research works in the area of corpus and computational linguistics are now data-driven. When dealing with data, there is a need to check sentences for variations and inconsistencies. Style and grammar checkers can be used for this purpose. However, recent technologies rely on manually developing rules, which is a time-consuming process and a herculean task. In this paper, a statistics-based rule generation framework – that can be used to learn spelling variations, affix usage, and common mistakes made – is presented. As domain, this research is focused on the Filipino language, characterized as a language with high degree of inflection. Monolingual corpora, annotated documents, as well as a tagged data were collected. The monolingual corpus was modeled and machine learning was used to aid in detecting spelling variations; the tagged data was processed and data association was applied to determine affix usage; and a subset of the annotated documents was digitized and used as training data for a statistical machine translation engine to determine common mistakes made. A total of 396 variant pairs, 16 affix usage, and 22 phrase pairs were generated and transformed into rules. A subset of these linguistic phenomena was reported in the literature, an indication that the framework can be used to automate linguistic tasks. The proposed variant scoring matches the style proposed by Sentro ng Wikang Filipino (SWF) with 30% recall and matches the style proposed by the Komisyon sa Wikang (KWF) Filipino with 60% recall, an indication that the style proposed by KWF is more inclined with the variant scoring. As future work, a policy paper could be drafted in coordination with experts in language planning.

Keywords: grammar checking, rule-based, statistical, corpus-linguistics, computational linguistics, natural language processing

Table of Contents

List of Tables

List of Figures

List of Listings

List of Equations

1 Research Description

This chapter discusses the overview of the current state of technology, the research objectives, scope and limitations of the research, significance of the research, research methodology, and research activities.

1.1 Overview of the Current State of Technology

Current technologies, both in natural language processing and in other areas of artificial intelligence, are now data-centric. Natural language processing and computational linguistics works both in and outside the country (Klein and Manning, 2003; Zuraw, 2006) require large amounts of documents. Research works in the Philippines include AutoCor (Dimalen and Roxas, 2007), Bantay-Wika (Ilao, Guevara, Llenaresas, Narvaez, and Peregrino, 2011), ICE-PHI (Bautista, Lising, and Dayag, 2004), PALITO (Dita et al., 2009; Roxas, Cheng, and Lim, 2009), and code-switching point detection (Oco and Roxas, 2012). AutoCor applied model-based language identification (LID) to categorize documents and build a corpus. It used large volumes of text, approximately 4,000 documents, as training data. On the other hand, Bantay-Wika is an example of culturomics; it tracked language change over time using approximately 61,000 tabloid articles as training data. ICE-PHI or the Philippine component of the International Corpus of English is a repository of collected documents and transcribed audio recordings. It contains at least one million words that were manually annotated. Another project, PALITO, is a repository of literary and religious texts, covering eight Philippine languages. Each language has approximately 250,000 words collected using manual means. Lastly, a system that can detect code-switching points in a sentence has been developed (Oco and Roxas, 2012). A combination of English and Tagalog Wikipedia articles covering at least 13 million words was used for language modeling.

Research works now rely heavily on collected and encoded data. In addition, text processing tools, e.g., code-switching point detection system (Oco, Wong, Ilao, Roxas, 2013a), POS tagger (Rabo and Cheng, 2006), require text inputs to be consistent, both in style and grammar, in order to function properly. In this regard, there is a need to ensure the quality of documents and text inputs, i.e., the style and grammar should be consistent. How can computers be used to perform quality assessment on data? Style and grammar checkers, programs that can perform grammar checking, can do this. Grammar refers to the scientific aspect of the study of language – focusing on syntax and morphology – while style refers to the matters that distinguish writing variations and consistencies (Lynch, 2008). Style and grammar checking can then be described as the process of (1) detecting if there is style and grammar variant or inconsistency in the sentence; (2) locating exactly where the variant or inconsistency is; (3) determining the type of variation or inconsistency; (4) notifying the user about it; and (5) suggesting ways to address it with possible linguistic explanations. Grammar checkers are often used in text editing but can also be used in optical character recognition, speech recognition, and machine translation (Alam, UzZaman, and Khan, 2006).

Approaches in grammar checking are categorized into three (Naber, 2003; Oco and Borra, 2011): (1) parser-based, (2) statistics-based, and (3) rule-based. In the country, several thesis and research projects worked on sentence analysis and grammar checking: (1) A semantic analyzer that has the capability to check syntax and semantic relationships in a Tagalog sentence has been developed (Ang, Cagalingan, Tan, and Tan, 2002); (2) a number of studies (Jasa, Palisoc, and Villa, 2007; Dimalen and Dimalen, 2007) on the other hand focused on developing parser-based Filipino grammar checker extensions for OpenOffice.Org Writer; and in a recent work, (3) the Tagalog component of LanguageTool – a rule-based style and grammar checker – was developed.

Parser-based checking utilizes parsers and parsing techniques, an input is accepted if parsing succeeds, and inconsistencies are detected if parsing or unification fails. This approach is grammar-dependent; a robust and complete grammar that covers all types of sentences is needed to ensure precision and recall. Complete grammars are hard to build and require expert knowledge. Parser-based Tagalog grammar checkers and sentence analyzers (Jasa et al., 2007; Dimalen and Dimalen, 2007; Ang et al., 2002) do not perform well outside the coverage of the grammar. Parser-based grammar checkers often have low precision rates because of incomplete grammar. Input not covered by the grammar cannot be detected.

Statistics-based checking utilizes POS-annotated corpus. Probability scores are computed to determine the statistical percentage of a sentence occurring. If a sentence has a low probability score, there is a high chance that it contains inconsistencies. Statistics-based grammar checkers require a flexible corpus in which they were trained. However, an expert in the field (Mark Johnson, email, February 01, 2011) added that "it is difficult to tell where it is and how it should be fixed".

Rule-based grammar checkers (Naber, 2003; Oco and Borra, 2011), on the other hand, can pinpoint exactly where the inconsistencies are and offer suggestions on how to fix them. This approach is more feasible given limited resources. Rule-based checking utilizes a set of rules, manually built incrementally, which it matches against an input. It offers certain advantages compared to other types of grammar checkers; rules are easy to configure and can adjust to the user's needs. With rule-based grammar checking, "the patterns being captured are sentences with inconsistencies" (Oco and Borra, 2011). This makes rule-based grammar checkers dependent on the rules declared for error checking coverage.

Rule-based approaches normally involve manual construction of rules. It has been mentioned in a related study (Konchady, 2009) that rule-based systems suffer from low recall because if there are fewer rules, fewer errors are detected. There is a need to generate more rules to broaden the error-checking coverage. Current approaches rely on transforming reference grammar books and expert knowledge to machine-readable rules. This is time-consuming and most of the rules generated are not commonly committed by native speakers. There is a need to focus on areas with variation and the most common mistakes by analyzing written data. Analysis will yield statistical data, which can be transformed to rules.

This research aims to answer this question: how can computers be used to extract rules from training data? This can be addressed through the development of a statistical rule generation framework; areas with variations and common mistakes are learned, rules are generated faster, and error-checking coverage is broadened.

1.2 Research Objectives

This section discusses the general objective and the specific objectives of the research.

1.2.1 General Objective

To develop a statistical rule generation framework for a rule-based style and grammar checker for Filipino.

1.2.2 Specific Objectives

This research aims to:

1.2.2.1 collect monolingual and parallel corpora using manual and automatic means;
1.2.2.2 computationally represent, model, and extract the features of the domain language;
1.2.2.3 categorize different style and grammar errors;
1.2.2.4 determine areas with variations in the domain language and provide statistical measures;
1.2.2.5 determine common mistakes by analyzing parallel corpora;
1.2.2.6 generate the rules; and
1.2.2.7 evaluate the system using standard metrics.

1.3 Scope and Limitations of the Research

This research will focus on the variety of the Filipino language with grammatical properties identical to Tagalog. Monolingual corpora can be collected from Wikipedia XML dumps, RSS feeds, and other sources. These can be used to model the language and determine areas with variations, e.g., spelling variation, affixation variation. Similarity measures (e.g., Dice's Coefficient, Out-of-Place measure) can be used both as a feature and to measure if enough data have been collected – increasing the size of the corpora does not significantly increase the similarity measure above a particular threshold (Oco, Ilao, Roxas, and Syliongka, 2013b). Available training data such as student submissions, especially those with marked errors and annotations by a faculty member from the Filipino department, can be used for the parallel corpus. Statistical machine translation concepts can be applied to learn the common mistakes made.

Sentences involving idiomatic expressions, interjections, sayings, and quotes will not be covered. Intra-word code-switching involving phonetic reduplication will also not be covered. However, this research will include sentence types covered by the training data that will be used: declarative, interrogative, exclamatory, and imperative.

The domain language can be modeled in terms of the following: character n-gram, word n-gram, word position, and POS tags. Also, existing Filipino tagsets (Rabo and Cheng, 2006; Miguel and Roxas, 2007; Manguilimotan and Matsumoto, 2011; Oco and Borra, 2011; Oco and Roxas, 2012) can serve as basis for the new tagset. The tagset that will be developed will be used to increase the chances of identifying and classifying errors.

Aside from checking the literature, manual bootstrapping will be applied to categorize errors. This involves manual analysis of grammar books to determine which features will be extracted. This research will cover style and grammar errors, but will not cover fully semantic errors. Style and grammar errors are characterized as patterns where a word should be replaced or a group of words should be transposed.

LanguageTool, an existing rule-based style and grammar checker engine (Naber, 2003; Oco and Borra, 2011) can be utilized. It primarily requires rules stored in xml files to properly function. The research will focus on generating rules, more specifically, rule templates. The rules generated from this research can be added to LanguageTool.

The system will be evaluated in terms of standard metrics used in the literature (Jasa et al., 2007; Oco and Borra, 2011): precision, recall, and f-measure. Accuracy will also be used to measure the number of properly identified error-free sentences.

1.4 Significance of the Research

It has been stated in the literature (Jurafski and Martin, 2000) that it is important to show how language-related algorithms and techniques can be applied to important real-world problems: spelling checking, text document search, speech recognition, web-page processing, part-of-speech tagging, machine translation, spoken-language dialogue agents, and the like. This research aims to implement a rule-based style and grammar checker for Filipino based on a statistical rule generation framework.

Tagalog is the basis for the Filipino language, the official language of the Philippines. According to the latest NSO data (Roxas, Lim, and Cheng, 2009), there are 22,000,000 native speakers of Tagalog as of year 2000. This makes it the highest in the country, followed by Cebuano with 20,000,000 native speakers. Tagalog is very rich in morphology; Tagalog words are normally composed of root words and affixes (Ramos, 1971) and a language with "high degree of inflection" (Dimalen and Dimalen, 2007). This research could serve as basis for future researches with regard to the computational aspect (i.e., language technology) of the Filipino language.

The existence of style and grammar checkers are not only useful in text editing, but can also be applied in optical character recognition, speech recognition, and machine translation, as stated in literature (Alam et al., 2006). The versatility of such tool makes it indispensable and one of the most widely used tools within natural language processing.

The development of a rule-generation framework will generate rules faster and could broaden error-checking coverage. False positives could also be avoided by just focusing on the common mistakes people make.

1.5 Research Methodology

The research methodology, shown in Figure 1-1, is divided into three: (1) data collection, (2) extraction, and (3) rule development. These methods are also tailored to include other development tasks.

Figure 1-1. Research Methodology

Data collection refers to the process of gathering documents. Related literature are also studied and analyzed. Extraction involves computationally representing the different data collected and extracting important features. Once areas with variations and common mistakes are identified, and converted into rules.

2 Review of Related Literature

This chapter gives a review of related literature. Topics involving corpus building and analysis, style and grammar checking, POS tagging, and data mining were studied.

2.1 Corpus Collection, Building, and Analysis

A corpus is a large collection of data. Corpus building, as the name suggests, is the process of building a corpus. In the country, corpus linguistics works, such as AutoCor (Dimalen and Roxas, 2007), Bantay-Wika (Ilao et al., 2011), ICE-PHI (Bautista et al., 2004), PALITO (Dita et al., 2009), and code-switching point detection (Oco and Roxas, 2012), used large amounts of data.

AutoCor is a system used for automatic corpus building. Web crawlers were used to mine the internet for documents and applied model-based language identification (LID) to categorize these documents. It initially used small volumes of text as training data and used bootstrapping in the language identification task; gradually increasing the training data with each identified document. The final corpus contains 4,000 documents. One of the problems identified is the low recall when closely-related languages are involved, which was addressed by modeling the unique words of the languages into character trigrams and using these instead.

Another project is Bantay-Wika, which is an example of culturomics. It tracked language change over time using approximately 61,000 tabloid articles as training data. One output of the Bantay-Wika project is the development of computational models that aided in tracking language change. Just like AutoCor, a web crawler was used to mine data from the web. However, there were certain dates that did not have any downloadable documents. The documents were processed, language features were extracted and, together with manual analysis, competing word forms were determined.

ICE-PHI or Philippine component of the International Corpus of English is a repository of collected documents and transcribed audio recordings. Unlike AutoCor and Bantay-Wika, the ICE-PHI project relied on human encoders and annotators. It contains at least one million manually-annotated words. The manual process exposed ICE-PHI to different human errors, e.g., unclosed tag (Davis Dimalen, email, 2011).

PALITO is a repository of literary and religious texts covering eight Philippine languages – Bikol, Cebuano, Hiligaynon, Ilocano, Kapampangan, Pangasinense, Tagaloy, and Waray. Just like ICE-PHI, human encoders were recruited or hired to come up with at least 250,000 words per language. News articles, short stories, and bible verses were manually encoded, and several informants were hired to verify the encoding. Its main objective is to allow researchers to manually annotate text documents and to allow an expert to verify these annotations.

A system (Oco and Roxas, 2012) that can detect code-switching (CS) points in a sentence was studied. Sentences from ICE-PHI were manually analyzed to describe the behavior of English-Tagalog CS. Just like AutoCor, character n-gram was used to perform LID. However, word n-gram was also utilized to address the problem of interlingual homographs, i.e., words that exist in more than one language. As training data for the language models, Wikipedia articles were used; ten million words from the English Wikipedia and three million words from the Tagalog Wikipedia.

Documentation efforts include the use of Linguist's Assistant (LA) to document Tagalog (Castilo, Go, Lam, Syson, Xu, Ong, and Beale, 2014). It is a computational tool, which requires manual analysis and encoding, to describe languages. It was noted that several complications in the Tagalog verb are not easily handled by the LA. Aside from this, LA was also used in translation and transforming corpora into rules (Allman, Beale, and Richard Denton, 2014).

It was noted that automatic collection (Dimalen and Roxas, 2007; Ilao et al., 2011) of text documents yield higher word count than using manual means (Bautista et al., 2004; Dita et al., 2009). Also, large amounts of data are freely available online (Oco and Roxas, 2012) and can be downloaded in machine-readable format. Finally, available tools (Beale et al., 2012) can aid in the manual aspects of corpus building and analysis.

2.2 Style and Grammar Checking

Grammar refers to the scientific aspect of the study of language – focusing on syntax and morphology – while style refers to the matters that distinguish writing variations and consistencies (Lynch, 2008). Style and grammar checkers, programs used for style and grammar checking, detect inconsistencies in an input (Naber, 2003). An expert in the field (Mark Johnson, email, February 01, 2011) added, that grammar checkers “should also propose a correction and tell exactly where the inconsistency is, and how it should be fixed”. Style and grammar checking can then be described as the process of (1) detecting if there is style and grammar variation or inconsistency in the sentence; (2) locating exactly where the variant or inconsistency is; (3) determining the type of variation or inconsistency; (4) notifying the user about it; and (5) suggesting ways to address it with possible linguistic explanations.

Three types of grammar checkers have been enumerated in a related study (Naber, 2003) – parser-based checking, statistics-based checking, and rule-based checking.

2.2.1 Parser-based

Parser-based checking utilizes parsers and parsing techniques. A parse tree is developed and if parsing fails, it can be assumed that the sentence contains inconsistencies. This approach is grammar-dependent; a robust and complete grammar that covers all types of variations and inconsistencies is needed to ensure accuracy. In the country, several thesis and research projects worked on parser-based approaches: (1) A semantic analyzer that has the capability to check syntax and semantic relationships in a Tagalog sentence has been developed (Ang, Cagalingan, Tan, and Tan, 2002); (2) a number of studies (Jasa, Palisoc, and Villa, 2007; Dimalen and Dimalen, 2007) on the other hand focused on developing parser-based Filipino grammar checker extensions for OpenOffice.Org Writer; and in a recent work, (3) the Tagalog component of LanguageTool – a rule-based style and grammar checker – was developed. However, these systems do not perform well outside the coverage of the grammar. Parser-based grammar checkers often have low precision rates because of incomplete grammar.

2.2.2 Statistics-based

Statistics-based checking utilizes POS-annotated corpus. Probability scores are computed to determine the statistical percentage of a sentence occurring. High probability score can denote consistency and low probability scores can denote inconsistencies. Probabilistic context free grammar or PCFG (Klein & Manning, 2003) can be utilized. Statistic-based grammar checkers require a flexible corpus in which they were trained. Also, “it is difficult to tell where the inconsistency is and how it should be fixed” (Mark Johnson, email, February 01, 2011).

2.2.3 Rule-based

Rule-based checking utilizes a set of rules which it matches against an input. If a pattern matches a certain rule, a variant or inconsistency exists. Rule-based grammar checking offers certain advantages compare to other types of grammar checkers – parser-based grammar checkers and statistic-based grammar checkers. Parser-based grammar checkers require an extensively-written grammar to properly work while statistic-based grammar checkers require a flexible corpus in which they are trained. Rule-based grammar checkers can pinpoint exactly where the variants and inconsistencies are and offer suggestions on how to address them. Aside from building the rules incrementally, rules are easy to configure and can adjust to the user’s needs.

Two types or rule-based grammar checkers were identified in a related study (Konchady, 2009). These are automatic-based system and manual-based system. Automatic rule-based systems have automatically generated rule sets that have “reasonable accuracy”. Manual rule-based systems are grammar checkers whose rules were manually created. This method offers users with very “descriptive and appropriate suggestions to correct errors”.

An advantage of manual-based system over an automatic-based system is as follows (Manu Konchady, email, May 08, 2011):

> An automatic system will create rules based on statistics in a tagged corpus. However, the tagged corpus may not cover all possible instances of tag patterns and therefore, the automatic rules may not generate all possible language POS tag patterns.

An example of a rule-based grammar checker that uses manual-based rule creation is LanguageTool (Naber, 2003). LanguageTool (LT) is an open-source style and grammar checker. It is also a plugin for OpenOffice.org and LibreOffice. Currently, LanguageTool supports different languages to a certain degree. The system takes a text input and produces a list of style and grammar variations and inconsistencies, and suggestions as output. It needs two language resources: the tagger dictionary and the rule file. Each word in the input is assigned a POS tag based on the declarations in the tagger dictionary. The words or phrases are then checked against a pre-defined xml rule file for errors. The xml rule file identifies errors as “patterns of words, part-of-speech tags, and chunks” (Oco and Borra, 2011).

There are two drawbacks in LanguageTool. One of the drawbacks is the manual creation and maintenance of grammar rules. It is a “tedious” process to maintain several hundreds of rule files and different languages; each language requires a different set of rule files. The presence of a community, working together in collaboration to maintain and process large amount of extensive grammar rule files, simplifies this drawback. Another drawback identified is the low recall rate of LanguageTool, because of the large number of patterns to be covered, the available rules cannot detect all these (Manu Konchady, email, May 08, 2011).

LanguageTool supports 29 languages. These include Chinese, French, and Esperanto. These languages have characteristics different from Filipino. Chinese does not have word segmentation so the language maintainer handling the Chinese component introduced a lot of skip elements in the rule file. This process involves skipping certain characters until a particular set of tokens is seen. French on the other hand has gender attributes. This means that the gender of the verb and the adjective should also agree with the gender of the noun. The language maintainer handling the French component introduced gender in the French tagset. Esperanto is a constructed international auxiliary language. It was developed at the end of the nineteenth century (Bergen, 2001) and contains vocabulary from Romance and Germanic languages, and phonology from Slavic languages. Just like French, Esperanto also has gender attributes. However, most terms are masculine by default.

One characteristic the Filipino language has that is not present in these languages is syllable reduplication and the embedded thematic roles in verbs. Filipino has a high degree of inflection (Dimalen and Roxas, 2007) and high variation index (Ilao et al., 2011).

As most of these language components follow manual-based rule generation – often deriving data from expert knowledge, learner corpora, or reference grammar books – this research will follow a semi-automatic rule generation approach using statistics. This framework has never been fully realized in the LanguageTool community.

2.3 Rule-based Systems

Rule generation is also applied in other areas of NLP, particularly in chatterbot, named entity recognition (NER), and machine translation.

Artificial Linguistic Internet Computer Entity (ALICE) is an example of a chatterbot (Schumaker and Chen, 2010). It uses rules stored in XML files to generate responses. These responses are normally manually populated. To date, several versions of ALICE, each with its own personality, have been developed. One research work (Chantawotrong, 2006) focused on automatically developing the rule file by constructing a conditional frequency distribution (CFD) of triggers and responses using a chat corpus as training data. This approach involves calculating which keyword triggered a particular response. The Natural Language Toolkit (website: http://nltk.org/) was used to calculate the CFD.

General Architecture for Text Engineering (GATE) is a framework and graphical development environment to deploy language engineering components (Cunningham, Maynard, Bontcheva, and Tablan, 2002). One use (Wang, Li, Bontcheva, Cunningham, and Wang, 2006) of Gate, in combination with support vector machines (SVM), is to automatically extract hierarchical relations from text. Plugins exist for machine learning in tools like Weka, Rasp, among others. This makes GATE useful for natural language processing tasks.

The Hybrid English-Filipino Machine Translation System (Roxas, Borra, Cheng, Lim, Ong, and Tan, 2008) is a DOST-funded project that combines both parser-based and rule-based machine translation approaches. The rule-based approach incorporates automatic extraction of templates using strict chunk alignment with splitting (SCAS) and common words filtering (CWF).

2.4 POS Tagging

Part-of-speech or POS is a lexical category that defines the function of words. The Tagalog POS is similar to the POS of the English language. Ten Tagalog parts of speech were identified in a study (Santos, 1939): *pantukoy* (article), *pangngalan* (noun), *panghalip* (pronoun), *pandiwa* (verb), *pandiwari* (participles), *pang-uri* (adjective), *pang-abay* (adverb), *pang-ukol* (preposition), *pangatnig* (conjunction), and *pandamdam* (interjection). An improvement was proposed (Ramos, 1971) and added *pang-angkop* (ligatures) in the list. Part-of-speech tagging or POST is the process of labeling words in a text or in a corpus with a particular POS (Miguel and Roxas, 2007). POS Tagging is essential in areas of translation, grammar checking, and language generation. The list of POS used to label words is called a tagset.

2.4.1 Structured Tagsets

To provide additional structure to a tagset, additional attributes can be added. An example in Filipino is the attribute *plurality*, which could either be *singular* or *plural* (e.g., *maganda* vs. *magaganda* /beautiful/). Structured tagsets can be categorized into two: positional tagsets and compact tagsets (Feldman and Hana, 2010). Positional tagsets are defined as a structured tagset composed of "tags coming from smaller atomic tagsets associated with a particular morpho-syntactic property", often encoded with one or more attributes. Table 2-1 shows an example set of attributes of the Russian tagset (Feldman and Hana, 2010). Compact tagsets are similar to positional tagsets. In a positional tagset, all tags have the same length, encoding all the attributes distinguished by the tagset. Attributes not applicable for a particular word have a N/A value. In a compact tagset, the N/A values are simply left out.

Table 2-1. Positional tagset attributes

POS	Abbr	Name	No. of values
1	p	Part of speech	12
2	s	SubPOS (Detailed POS)	42
3	g	Gender	4
4	y	Animacy	3
5	n	Number	3
6	c	Case	7
7	f	Possessor’s gender	4
8	m	Possessor’s number	2
9	e	Person	4
10	r	Reflexivity	2
11	t	Tense	4
12	b	Verbal aspect	3
13	d	Degree of comparison	3
14	a	Negation	2
15	v	Voice	2
16	i	Variant, Abbreviation	7

2.4.2 Tagalog POS Taggers

In the country, several compact tagsets and POS taggers have been developed. One study (Rabo and Cheng, 2006) proposed a tagset compatible with Tagalog. The tagset is composed of 59 tags covering 10 major POS Tags. Noun has 3 tags, pronoun has 9, determiner has 4, conjunction has 4, verb has 7, adjective 6, adverb 9, preposition 1, cardinal 1, and punctuation 5. These were modified (Miguel and Roxas, 2007) to include verb focus, ligatures, and several conjunction tags, to name a few. Unlike in the Russian tagset, interjections and digits are included. However, the Tagalog tagsets do not have the following attributes: gender, person, and tense.

2.5 Data Mining

Data mining provides “approaches for the identification and discovery of non-trivial patterns and models hidden in large collections of data” (Atzmueller, 2012). Studies in the Philippines that have used data mining approaches include the classification of disaster-related tweets (Lam, Paner, Macatangay, and Delos Santos, 2014), using data association to mine named entities in Philippine arts domain (Syliongka and Oco, 2014), and clustering of languages (Oco, Sison-Buban, Syliongka, Roxas, and Ilao, 2014b).

2.6 Summary

This section gives a summary of the related literature discussed. Table 2-2 shows a summary of different corpus building projects, Table 2-3 shows a summary of different grammar checkers, Table 2-4 shows a summary of different rule-based systems, and Table 2-5 shows a summary of different POS Taggers.

Table 2-2. Summary of different corpus building projects

System	Purpose	Domain language	Genre	Data Size	Method for collecting data	Data analysis
AutoCor	Corpus Building	English, Tagalog, Cebuano, Bikol	General	4,000 documents	Automatic: Web crawling	Computational modeling
Bantay-Wika	Language Trend Analysis	Filipino	Tabloid	61,000 articles	Automatic: Web crawling	Computational modeling
ICE-PHI	Repository	Philippine English	General	One million words	Manual: transcription of video recording and encoding collected text	Manual annotation
PALITO	Repository	Bikol, Cebuano, Hiligaynon, Ilocano, Kapampangan, Pangasinense, Tagalog, Waray	Religious and Literary	250,000 words per language	Manual: Digitization of bible verses, news articles, literary texts	Manual annotation
Code-switching point detection	Text Processing	English, Tagalog, Taglish	General	English: ten million words Tagalog: three million words	Automatic: Wikipedia XML dumps	Computational modeling

Table 2-3. Summary of different grammar checkers

System	Approach	Detect	Locate	Determine the cause	Notify the user	Feedback	Problem
PanPam	Parser-based	Parsing or unification fails	Identify where parsing or unification failed	Knowing which stage failed	Separate window	Canned Text	Limited grammar
FiSSAn	Parser-based	Parsing or unification fails	Identify where parsing or unification failed	Knowing which stage failed	Textbox	None	Limited grammar
Plug-in for OpenOffice	Parser-based	Parsing or unification fails	Identify where parsing or unification failed	Knowing which stage failed	Underline	Unknown	Limited grammar
LT	Rule-based	Pattern-matching	Identify which words matched the rules	The type is specified in the rule	Underline	Explanation and suggestions	Limited rules

Table 2-4. Summary of different rule-based systems

System	Purpose	Algorithm applied to generate rules
ALICE	Chatterbot	Conditional frequency distribution
GATE	Graphical development environment for text processing applications	Support vector machine
Hybrid English-Filipino Machine Translation System	Machine translation	Strict chunk alignment with splitting and common words filtering

Table 2-5. Summary of different POS tagsets

Tagset	Domain Language	Type	Unique Attributes
Russian Positional Tagset	Russian	Positional Tagset	Has the following attributes: gender, person, tense
Rabo Tagset	Tagalog	Compact Tagset	Has the following POS: interjections and tense
Modified Rabo Tagset	Tagalog	Compact Tagset	Has the following POS: interjections and tense

Table 2-6. Summary of different data mining research works

Data Mining Approach	Algorithm	Data Type	Domain
Classification	Naive Bayes and SVM	Tweets	Disasters
Association	Association Rule Mining	Website Articles	Arts
Clustering	K-means Clustering	Trigram models	Religious and Literary

3 Theoretical Framework

This section discusses the theories and concepts that used in this research.

3.1 Language Modeling

A language model is a smaller representation of a language and is usually expressed in terms of character n-grams and their frequency count. A character n-gram is defined as an n-character slice of a word (Dimalen and Roxas, 2007). Deriving the formal definition of a longest common subsequence (Kondrak, 2005), the standard formulation for an n-gram is as follows: given a string $X = \{x_1 ... x_k\}$, character sequence $Z = \{z_1 ... z_n\}$ is an n-gram of X if there exist a strictly incrementing sequence $i_1 ... i_n$ of indices of X such that for all $j = 1 ... n$, $X_{i_j} = Z_j$.

For n of size one, it is called a unigram, size two is called a bigram, and size three is called a trigram. As an example, the list of trigrams that can be generated from the word "*kumuha*" (/got/) are {*_ku*,*kum*,*umu*,*muh*,*uha*,*ha_*}. An underscore signifies the beginning and end of a word and are part of the trigram. For $n \geq$ four, these are referred to by the value of n (e.g., 4-gram or four-gram). The number of possible combinations increases as n increases.

3.2 Language Identification

Language identification (LID) is the process of identifying which language a text input is in. It can enhance document mining tasks and is used in the areas of computational linguistics and corpus linguistics (Ilao et al., 2011; Dimalen and Roxas, 2007). LID can be mathematically described as the argmax function in Equation 3-1, where Ŀ is the identified language, X is the text input, Γ is the set of target languages, and $S(X,L)$ is the similarity score of X with language L. The process of performing LID involves the following: (1) gathering volumes of text as training data through automatic or manual means, (2) creating language models using the data collected, (3) using similarity measures to determine which among the set of languages the input is in. The language that yields the highest similarity measure is identified as the language of the input.

$$\dot{L} = \underset{L \in \Gamma}{argmax}\ S(X, L)$$

Equation 3-1. Language identification as an argmax function

In the field of computational linguistics, Yeong and Tan (2010) compared and analyzed 5 approaches to detect Malay and English words and phrases in a text document. These approaches are affixation information, vocabulary list, alphabet n-gram, grapheme n-gram, and syllable structure. Both affixation information approach and vocabulary list approach utilize dictionary look-ups. On the other hand, alphabet n-gram, grapheme n-gram, and syllable structure utilize language models to perform LID. It has been shown that dictionary look-up methods have lower accuracy rates than model-based methods.

Both model-based and dictionary-based approaches can be used in LID. Oco and Roxas (2012) utilized both to perform code-switching point detection. The concepts behind LID can also be used to perform other text processing and text categorization tasks.

3.3 Similarity Measures

The history of LID can be traced to similarity measures and edit distances, which compute how similar two strings are. String similarity metrics like the Normalized Levenshtein Distance (Levenshtein, 1965), Dice Similarity Coefficient (Dice, 1945; Oco, Syliongka, Ilao, and Roxas, 2013c), and Out-of-place measure (Cavnar and Trenkle, 1994) are often employed. The equation for Dice coefficient is shown in Equation 3-2, where X and Y represent distinct sets. On the other hand, as explained by Dimalen and Roxas (2007), the out-of-place measure (shown in Figure 3-1) determines how far an n-gram in the trigram model of the input (i.e., Document Profile) is from its place in the language model (i.e., Category Profile).

$$Dice\ Coefficient\ = 2X \cap Y / X + Y$$

Equation 3-2. Dice's similarity coefficient

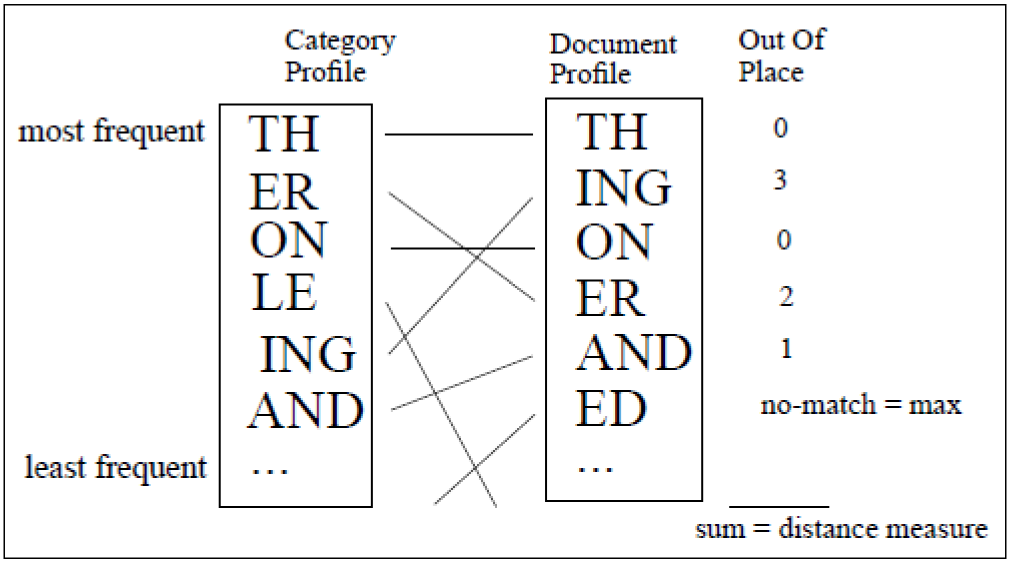

Figure 3-1. Out-of-place measure

3.4 Statistical Machine Translation

Machine translation is the process of translating one language to another with the aid of computers. One of the approaches is through statistical methods, called statistical machine translation (SMT). It utilizes a bilingual corpora or a parallel corpus and learns patterns from it. Its goal is to find the probability that a string *t* is the translation of a given string *s*, and the alignment *a* between the two. The probability distribution is shown in Equation 3-3. Finding the best translation is defined in the argmax function in Equation 3-4. The best translation is the one that yields the highest probability.

$$P(a, t|s)$$

Equation 3-3. Probability distribution of SMT

$$T = \underset{t \in W}{argmax}\ P(t|s)$$

Equation 3-4. SMT as an argmax function

Several translation models can be applied. One of them is expectation maximization (Dempster, Laird, and Rubin, 1977) or EM. Its goal is to find the maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. It is composed of a two-step iterative process: (1) compute expected counts for all word pairs, and (2) compute new maximum likelihood estimates from the expected counts.

3.5 Types of Errors

Several Filipino linguistic phenomena have been reported in the literature, which include spelling changes. These are shown in Table 3-1 and Table 3-2.

Table 3-1. Linguistic Phenomena (Zuraw, 2006)

Rule / Phenomenon	Examples
Intervocalic Tapping	Dumi > marumi
Vowel Height Alternations	Halo+in > haluin
Assimilation	Pang+butas = pambutas
Nasal Substitution	Pili > mamili
Syncope	Tingin+an = tignan
Partial Reduplication	Nag-pi-friendster
Infix location	Gumraduate vs. Grumaduate
Infix in vs. Prefix ni	Linuto vs. Niluto
Reduplication Location	Paglalagyan vs. Papaglagyan

Table 3-2. Linguistic phenomena (Sentro ng Wikang Filipino – Diliman, 2008)

Rule / Phenomenon	Examples	Variant	Wrong	Exception
Assimilation	Bukang-bibig Sangla Barong-barong Pang+paligo = pampaligo	Bukambibig Sanla Barumbarong	None	Tanglaw Singsing Mang-snatch
Intervocalic Tapping	Hiwaga raw Araw raw Na+dito > narito	None	Hiwaga daw Araw raw	Idaing Idemanda Idiin
Vowel Height Alternations	Pinto+an = pintuan Babae > Kababaihan	None	None	Sinehan Sira+an = siraan
Word Reduplication	Ano-ano Sari-sari	None	Anu-ano Sarisari	Haluhalo Salusalo
Syncope	Dakip+in = dakpin	None	None	None
Reduplication Location	Makauunawa	Makakaunawa	None	None
Partial Reduplication	Magba-brown out	None	Magbra-brown out	Magsha-shampoo
Spelling variation	Bituin Kabiyak Kapuwa	Bitwin Kabyak Kapwa	None	None
Dash	Taga-Cebu Paki-average Pa-Luneta	None	Kay-sigla Kay-bagal	Paloob Tagaluto Makakaliwa
Verb plurality	Naggagandahan ang mga babae	None	Naggagandahan ang babae	None
Noun plurality	Ang mga painting	Ang paintings	Ang mga paintings	None
Ligature Usage	Pinagmasdan ni Abdulla na lumalakad ang 211 pamilya	None	Pinagmasdan ni Abdullang lumalakad ang 211ng pamilya	None
Determiner removal	Lungsod Quezon	None	Lungsod ng Quezon	None
Ng vs. Nang	Kumain ng karne Kumain nang maayos	None	Kumain nang karne Kumain ng maayos	None

The International Tag Set (ITS) 2.0[1] implements a list of quality issue types. Lists are shown in Table 3-3, Table 3-4, Table 3-5, Table 3-6, and Table 3-7. The main purpose is to provide support for general and/or particularly common quality issues.

Table 3-3. Quality issue types

Value	Description	Examples
terminology	An incorrect term or a term from the wrong domain was used or terms are used inconsistently	• The localization had Pen Drive when corporate terminology specified that USB Stick was to be used; The localization inconsistently used Start and Begin.
mistranslation	The content of the target mistranslates the content of the source	• The English source reads "An ape succeeded in grasping a banana lying outside its cage with the help of a stick" but the Italian translation reads "l'ape riuscì a prendere la banana posta tuori dall sua gabbia aiutandosi con un bastone" ("A *bee* succeeded…")
omission	Necessary text has been omitted from the localization or source	• One or more segments found in the source that should have been translated are missing in the target
untranslated	Content that should have been translated was left untranslated	• The source segment reads "The Professor said to Smith that he would hear from his lawyer" but the Hungarian localization reads "A professzor azt modta Smithnek, hogy he would hear from his lawyer."
addition	The translated text contains inappropriate additions	• The translated text contains a note from the translator to himself to look up a term; the note should have been deleted but was not.

[1] http://www.w3.org/TR/its20/

Table 3-4. Quality issue types

duplication	Content has been duplicated improperly	• A section of the target text was inadvertently copied twice in a copy and paste operation.
inconsistency	The text is inconsistent with itself (NB: not for use with terminology inconsistency)	• The text states that an event happened in 1912 in one location but in another states that it happened in 1812.
grammar	The text contains a grammatical error (including errors of syntax and morphology)	• The text reads "The guidelines says that users should use a static grounding strap."
legal	The text is legally problematic (e.g., it is specific to the wrong legal system)	• The localized text is intended for use in Thailand but includes U.S. regulatory notices. • A text translated into German contains comparative advertising claims that are not allowed by German law
register	The text is written in the wrong linguistic register of uses slang or other language variants inappropriate to the text	• A financia text translated into U.S. English refers to dollars as "bucks".

Table 3-5. Quality issue types

locale-specific-content	The localization contains content that does not apply to the locale for which it was prepared	• A text translated for the Japanese market contains call center numbers in Texas and refers to special offers available only in the U.S.
locale-violation	Text violates norms for the intended locale	• A text localized into German has dates in YYYY-MM-DD format instead of in DD.MM.YYYY • A translated text uses American-style foot and inch measurements instead of centimeters.
style	The text contains stylistic errors	• Company style dictates that all individuals be referred to as Mr. or Ms. with a family name, but the text refers to “Jack Smith”.
characters	The text contains characters that are garbled or incorrect or that are not used in the language in which the content appears	• the text should have a • but instead has a ¥ sign • A text translated into German omits the umlauts over ü, ö, and ä • A Japanese localization contains characters like మ and ఊ (from Telugu)
misspelling	The text contains a misspelling	• A German text misspells the word *Zustellung* as *Zustellüng*

Table 3-6. Quality issue types

typographical	The text has typographical errors such as omitted/incorrect punctuation, incorrect capitalization, etc.	• An English localization has the following sentence: *The man whom, we saw, was in the Military and carried it's insignias*
formatting	The text is formatted incorrectly	• Warnings in the target text are supposed to be set in italic face, but instead appear in bold face • Margins of the text are narrower than specified
inconsistent-entities	The source and target text contain different named entities (dates, times, place names, individual names, etc.)	• The name *Thaddeus Cahill* appears in an English source but is rendered as *Tamaš Cahill* in the Czech version • The date February 9, 2007 appears in the source but the translated text has "2. September 2007."
numbers	Numbers are inconsistent between source and target	• The source text states that an object is 120 cm long, but the target text says it is 129 cm. long.
markup	There is an issue related to markup or a mismatch in markup between source and target	• The source segment has five markup tags but the target has only two • An opening tag in the localization is missing a closing tag

Table 3-7. Quality issue types

whitespace	There is a mismatch in whitespace between source and target content	• A source segment starts with six space characters but the corresponding target segment has two non-breaking spaces at the start.
pattern-problem	The text fails to match a pattern that defines allowable content (or matches one that defines non-allowable content)	• The quality checking tool disallows the regular expression pattern ['"”’][\.,] but the translated text contains A leading “expert”, a political hack, claimed otherwise.
internationalization	There is an error related to the internationalization of content	• A line of programming code has embedded language-specific strings • A user interface element leaves no room for text expansion • A form allows only for U.S.-style postal addresses and expects five digit U.S. ZIP codes
length	There is a significant difference in source and target length	• The translation of a segment is five times as long as the source
other	Any issue that cannot be assigned to any values listed above.	
uncategorized	The issue has not been categorized	• A new version of a tool returns information on an issue that has not been previously checked and that is not yet classified

4 Statistics Based Rule Generation Framework

This chapter discusses the statistics-based rule generation framework, shown in Figure 4-1, in relation to the research methodology.

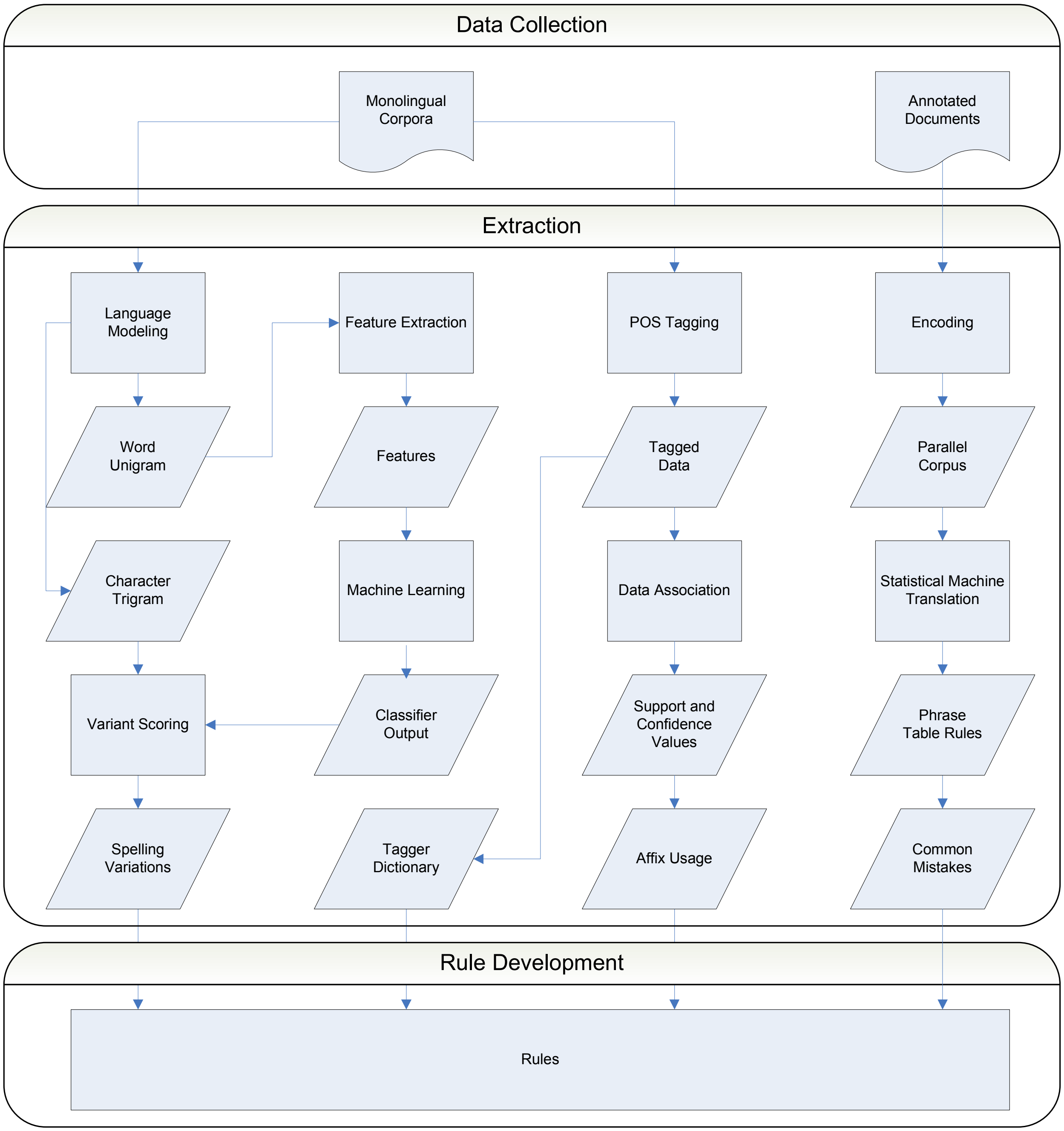

Figure 4-1. Statistics-based rule generation framework

4.1 Data Collection

Monolingual corpus from various sources and annotated documents are collected.

4.2 Extraction

The monolingual corpus undergoes language modeling to generate two language models: (1) a word unigram model and (2) a character trigram model. The word unigram model is then used in feature extraction to generate a feature set. The features undergo machine learning and the classifier output and the character trigram model, in tandem with variant scoring, is used to determine spelling variations. The corpus also undergoes POS tagging to generate tagged data. However, if a tagged data is already available, this step is skipped and affix usage is derived using data association. Lastly, student submissions are encoded into a parallel corpus and used as training data in a statistical machine translation (SMT) engine. This generates phrase table rules which are then used to describe common mistakes.

4.2.1 Language Modeling

Word unigram and character n-gram models of the corpus are generated using Apache Nutch[2] and the Stanford Research Institute Language Modeling toolkit[3] (SRILM), respectively. Listing 4-1 shows sample word unigrams and Listing 4-2 shows sample character trigrams. The language models are in this form: <n-gram> <frequency>.

```
sa      2,028,377
<s>     1,782,928
</s>    1,782,928
ng      1,626,249
,       1,577,049
ang     1,575,838
.       1,568,330
mga     673,087
ay      545,496
ni      323,814
```

Listing 4-1. Top 10 Tagalog word unigrams

```
ng_     7,675,177
ang     4,575,086
_na     3,083,907
_sa     2,643,366
sa_     2,388,606
_an     2,027,396
_ma     1,975,713
_ng     1,842,424
_pa     1,811,321
an_     1,803,874
```

Listing 4-2. Top 10 Tagalog trigrams

4.2.2 Feature Extraction

Word pairs are taken from the language model and a computer program is used to extract the different features. Features considered for this research are the following: similarity measures, string length, character difference, character location and adjacent characters, and word frequency.

---

2 https://nutch.apache.org/
3 http://www.speech.sri.com/projects/srilm/

4.2.3 Machine Learning

A subset of the feature set is annotated and machine learning is used to determine which features contribute to spelling variants. The Waikato Environment for Knowledge Analysis[4] (WEKA) is used.

4.2.4 Variant Scoring

Part of the definition of a grammar checker is to also propose a suggestion. To determine which spelling variant represents the language model more, a scoring mechanism from a related literature (Oco, Syliongka, Ilao, and Roxas, 2014a) – weighted score (WS) – was taken and modified. The modified equation is shown in Equation 4-1. Half of the score is percent frequency (PF), which is taken from the frequency count of the word in percent (i.e., frequency count of the word divided by the sum of the frequency count of both variants). The other half – also in percent – is frequency count of the trigram (PT) involved with the character difference. Each variant is scored and the variant with the higher MWS is considered for suggestion.

$$MWS = PF * .5 + PT * .5$$

Equation 4-1. Modified weighted score

4.2.5 POS Tagging

Part-of-speech tagging (POS tagging or POST) is an entirely different problem and not within the scope of this research so this process is skipped and a tagged data from a different source (Manguilimotan and Matsumoto, 2011) – consisting of words from an excerpt about Jose Rizal – is instead used. Sample word declarations are shown in Listing 4-3. It follows this format: <surface form of the word / word> <root> <prefix> <infix> <suffix> <reduplication> <POS Tag>.

```
magiging   magiging   _   _   _   _    VB-COAF
kukuha     kuha       _   _   _   ku   VB-COAF
magiging   magiging   _   _   _   _    VB-COAF
gagawa     gawa       _   _   _   ga   VB-COAF
uuwi       uwi        _   _   _   u    VB-COAF
pupunta    punta      _   _   _   pu   VB-COAF
papasok    pasok      _   _   _   pa   VB-COAF
babalik    balik      _   _   _   ba   VB-COAF
lalabas    labas      _   _   _   la   VB-COAF
```

Listing 4-3. Sample word declarations

Also, analysis is only limited to the actor focus (AF) due to the low number of supporting literature on the area of parts-of-speech and foci.

4.2.6 Data Association

The tagged data are treated as transaction pairs and data association is used to infer relations. The support and confidence values are generated and, in tandem with manual analysis, threshold values are set to identify affix usage.

4.2.7 Encoding and Statistical Machine Translation

The student submissions are encoded to produce a parallel corpus. Table 4-1 shows ten sample sentence pairs. The source text refers to sentences with errors and the translated text refers to the correct form. SMT is used to learn common patterns.

[4] http://www.cs.waikato.ac.nz/ml/weka/

Table 4-1. Sample sentences from the parallel corpus

Source Text	Translated Text
nagging	Naging
Napanalunan ni Court ang higit sa kalahating kaganapan sa Grand Slam.	Napanalunan ni Court ang higit sa kalahating torneo sa Grand Slam.
Nakuha rin ni Court and ikaunang ranggo noong 1973.	Nakuha rin ni Court and unang ranggo noong 1973.
rekord	record
Nanalo siya sa higit sa 100 na larong singles.	Nanalo siya nang higit sa 100 na larong singles.
labingisang	labing-isang
kwarter faynals	quarter finals
Noong sumunod na taon, natalo si Court sa huling laban niya kay Evonne Goolagong Cawley habang buntis sa una niyang anak na si Daniel, ipinanganak noong Marso 1972	Noong sumunod na taon, natalo si Court sa huling laban niya kay Evonne Goolagong Cawley habang buntis sa una niyang anak na si Daniel na ipinanganak noong Marso 1972
Isa si Court sa tatlong manlalaro na nakakamit ng "boxed set" na titulong Grand Slam.	Isa si Court sa tatlong manlalaro na nagkamit ng "boxed set" na titulong Grand Slam.
Siya rin ay natatangi sa pagkapanalo niya ng boxed set.	Natatangi rin siya dahil sa pagkapanalo niya ng boxed set.

4.3 Rule Generation

LanguageTool is a rule-based style and grammar checker engine. It uses two resources to work: (1) the tagger dictionary and the (2) rule file. It can run as an OpenOffice and LibreOffice extension or as a stand-alone program. Rules are generated as follows:

- spelling variants are taken from the results of the variant scoring;
- affix usage are taken from the support and confidence values; and
- common mistakes are taken from the phrase table rules.

4.3.1 Tagger Dictionary

The tagger dictionary is a text file that contains word declarations and their tag. Some examples are shown in Listing 4-4. The tagger dictionary follows this format: <token> <base form> <POS Tag>, where the base form is simply the token without the modifier linkers (i.e., “-ng” and “-g”). For this research, an earlier Tagalog tagger dictionary (Oco and Borra, 2011) was used and modified. It has a total of 7,849 entries.

```
Alemanyang    Alemanya    NPRO
Alfonso       Alfonso     NPRO
alikabok      alikabok    NCOM 2
alila         alila       NCOM 2
alimango      alimango    NCOM 1
alimura       alimura     NCOM 2
alin          alin        PINP NU S
aling         alin        PINP NU S
aling         aling       PINP NU S
```

Listing 4-4. Sample word declarations

4.3.2 Rule File

The rules on the other hand are stored in an XML file, which contains the patterns to be matched. These patterns could be represented in terms of tokens, regular expressions, and/or POS tags. Listing 4-5 shows a sample rule file. Each rule has three basic elements: (1) the pattern to be matched, (2) the message / suggestion, and (3) examples.

```
<pattern case_sensitive="no" mark_from="0">
        <token>cake</token>
</pattern>
<message>Do you mean <suggestion>keyk</suggestion>?</message>
<short>Loan Words</short>
<example correction="keyk" type="incorrect">Kumain kami ng
        <marker>cake</marker> kagabi.
</example>
<example type="correct">Kumain kami ng <marker>keyk</marker> kagabi.</example>
```

Listing 4-5. Sample rule

LanguageTool checks documents as follows:

- it separates an input into sentences and separates each sentence into tokens;
- tokens are given their tag using the declarations in the tagger dictionary;
- the tokens, together with their tag, are matched against the rule file;
- if a pattern matches, the user is notified and feedback with possible linguistic explanation or suggestion is provided.

5 Results and Analyses

This chapter discusses the results and analyses of the data collection, extraction, and rule development.

5.1 Data Collection

The following monolingual corpora were collected: (1) the PALITO corpus (Dita et al., 2009), (2) Tagalog Wikipedia (Oco and Roxas, 2012), and the (3) UP DSP Corpus (Ilao et al., 2011). The size of each corpus is shown in Table 5-1. It has been discussed in a related study (Oco et al., 2014a) that 290K words are enough to represent a language. However, to achieve the optimum representativeness, the largest corpus – UP DSP Corpus – was used for this research.

Table 5-1. Corpus size

Corpus	Number of Words
PALITO	290K
Tagalog Wikipedia	3M
UP DSP Corpus	31M

Aside from monolingual corpora, marked and annotated student submissions from the Filipino department of De La Salle University were also collected. These were submitted by students as part of the requirements in translation studies and contain translated English Wikipedia articles. They have been checked by their professor and most of the errors have corrections.

5.2 Language Modeling

The word unigram model contains a total of 666,217 unique unigrams. Figure 5-1 shows a log scatter plot of the top 10,000. The x-axis refers to the rank while the y-axis refers to the frequency count. It can be noticed that the language model follows a power law, similar to a Zipfian distribution; the frequency count of a trigram is inversely proportional to its rank. The equation for the model is shown in Equation 5-1, where r is the rank, n is the frequency count, a is a value between 5.0 and 7.0, and b is 1. The graph also follows the Pareto principle, where 80% of the trigrams are in the top 20%.

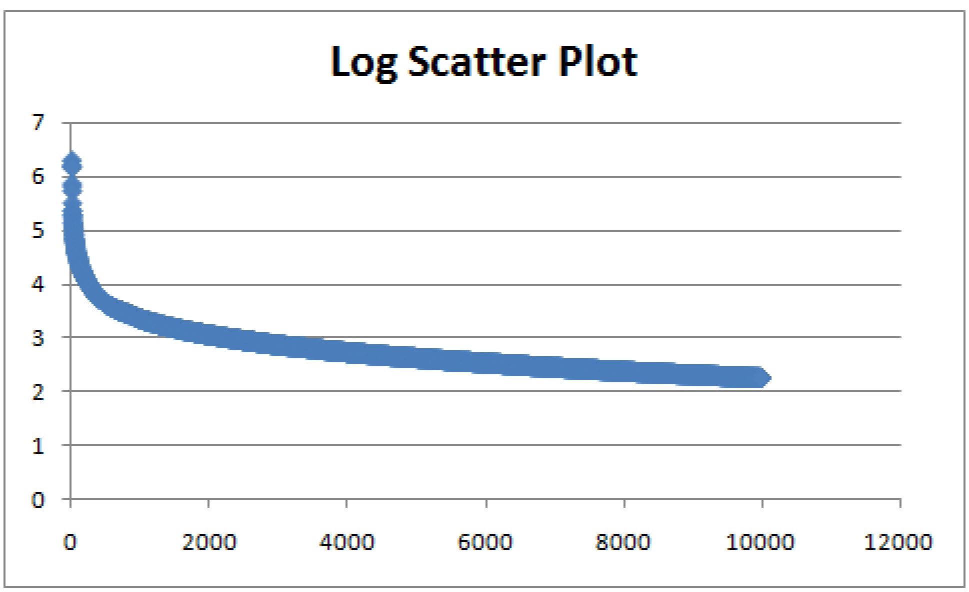

Figure 5-1. Log scatter plot of the frequency count

$$\log(r) = a - b\log(n)$$

Equation 5-1. Zipfian Model

Generating word pairs for all the unique trigrams and computing the Dice's coefficient and other features would be computationally expensive. A total of 221 billion instances would be generated. To address this issue, the language model was cleaned. The cleaning process is as follows:

- deleted unigrams with double quotes, commas, period, parenthesis, equal sign, digits, a capital letter, and unknown symbols;
- deleted unigrams beginning with a dash;
- deleted unigrams with less than 11 frequency counts;
- deleted unigrams with less than four characters.

The resulting language model, which is almost one tenth of the original, contains a total of 67,963 unique unigrams. Listing 5-1 shows the top 10 unigrams. The top 300 are shown in the appendix. It can be noticed that nouns and verbs are not present in the top 10.

```
hindi      231,660
isang      198,385
kung       166,616
niya       142,099
para       141,337
siya       139,178
nang       139,067
naman      129,414
lang       125,746
kanyang    121,829
```

Listing 5-1. Top 10 resulting word unigrams

For the character n-gram model, size three (i.e., trigrams) was used following LanguageTool (Naber, 2003) standards. Higher value of *n* would be computationally expensive and would cover a larger space, as seen in Table 5-2. Lower values, on the other hand, are not enough to represent the unique character sequences of a language, i.e., not enough to cover single-letter words (e.g., _y_). Trigrams offer a good combination of computational practicality and coverage. For this research, only the top 1,000 trigrams were used and those with low frequency counts were discarded, also following LanguageTool standards. Similar to a word unigram model, the trigram model also follows a power law and the Pareto principle, as shown in Figure 5-2.

Table 5-2. Number of possible combinations with respect to *n*

Size of *n*	**Possible Combinations[5]**
1	27^1
2	28^2
3 and above	$28^2 \times 27^{n-2}$

[5] The Philippine alphabet has 27 letters: the basic 26 letters and ñ.

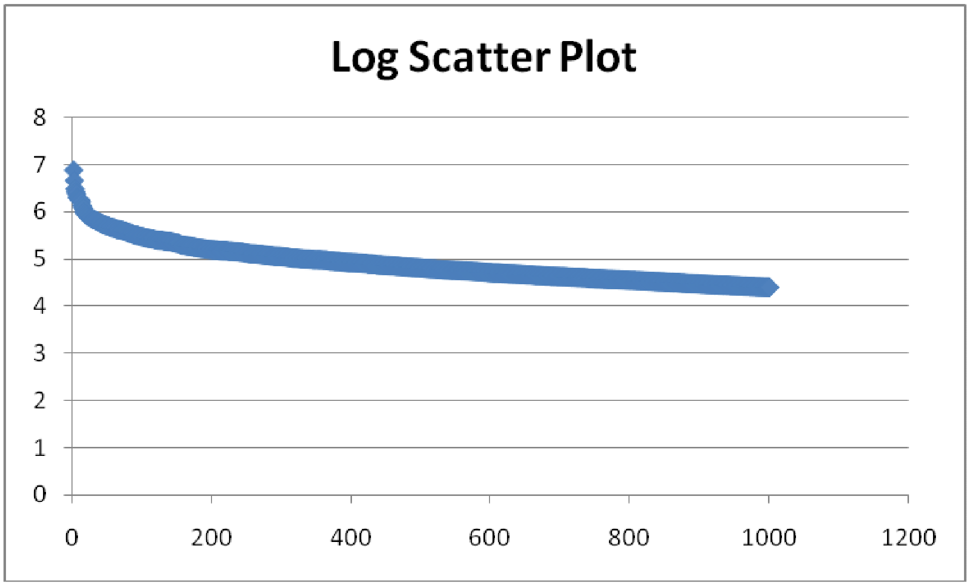

Figure 5-2. Log scatter plot of the frequency count

5.3 Feature Extraction

Based on the 67,963 unique unigrams, more than 2 billion instances can be generated for the feature set (i.e., the summation from 1 to 67,963). This would be computationally expensive for machine learning so thresholds were set when generating the features:

- word pairs with Dice's coefficient values is .85 and below are pruned – .85 was set as threshold because ocular inspection reveals minimal spelling variants below the .85 mark; and
- word pairs whose edit distance is greater than one are pruned – one was selected as most of the spelling changes reported in the literature involve single letters, both replacement and insertion/deletion.

Aside from Dice's coefficient and edit distance, other important features considered are: character difference (including indices and adjacent characters), frequency counts, and related values. To generate the character difference, wdiff[6] was used. However, wdiff only takes down the word difference and not the character difference. To solve this issue, each word was treated as a sentence and each character was treated as a word. Table 5-3 shows sample features; word pairs are "aabutan" vs. "aabutin" and "aawatin" vs. "aawitin". The character enclosed in [--] is replaced by the character enclosed in {++}. Also, a char type C would denote a consonant while a char type V would denote a vowel. Including those pruned due to bug errors from wdiff, a total of 7,454 instances were generated.

[6] https://www.gnu.org/software/wdiff/

Table 5-3. Sample features

Feature	Description	Example 1	Example 2
Word 1 (W1)	First word	aabutan	aawatin
Word 2 (W2)	Second word	aabutin	aawitin
W1 Length	String length	7	7
W2 Length		7	7
W1 space	Words with spaces	a a b u t a n	a a w a t i n
W2 space		a a b u t i n	a a w i t i n
Dice	Dice's coefficient	0.857143	0.857143
Edit	Edit Distance	1	1
Wdiff	Character difference	[-a-]{+i+}	[-a-]{+i+}
Wdiff Freq	Frequency count of the character difference	638	638
Wdiff Index	Index value of the character difference	6	4
Wdiff Position	Position in the word of thw wdiff: Start, Middle, Last	Middle	Middle
Char Left	Character on the left of the character difference	t	w
Char Right	Character on the right of the character difference	n	t
Char Left Type	Character type: Consonant (C) or Vowel (V)	C	C
Char Right Type		C	C
W1 Freq	Frequency count of the word	120	13
W2 Freq		584	60
W1 n-gram	Trigram with the character difference in the middle	tan	wat
W2 n-gram		tin	wit

5.4 Machine Learning

The goal of machine learning is to determine the features that constitute a spelling variant.

5.4.1 Classification

The features discussed in the previous section were simplified by removing:

- Redundant features (e.g., W1, W2, W1 space, W2 space);
- Features with the same value/range for all entries (e.g., Dice, Edit); and
- Superfluous features (e.g. Wdiff Freq, W1 Freq).

A variety of classification techniques were then applied using the default setting. A total of 575 random instances were manually annotated (i.e., variant or not) for this purpose. Listing 5-2, Listing 5-3, and Listing 5-4 show the confusion matrix for J48, Naive Bayes, and multilayer perceptron, respectively. The results for J48 indicate that no definite set of features fully indicate a spelling variant while the results for Naive Bayes and multilayer perceptron indicate that the features used were appropriate in detecting spelling variations. Among the three, multilayer perceptron showed the most promising results.

```
=== Summary ===

Correctly Classified Instances         460               80      %
Incorrectly Classified Instances       115               20      %
Kappa statistic                          0
Mean absolute error                      0.32
Root mean squared error                  0.4
Relative absolute error                 99.8054 %
Root relative squared error             99.9997 %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)     100      %
Total Number of Instances              575

=== Confusion Matrix ===

   a   b   <-- classified as
 460   0 |   a = not
 115   0 |   b = variant
```

Listing 5-2. Confusion matrix for J48

```
=== Summary ===

Correctly Classified Instances         516               89.7391 %
Incorrectly Classified Instances        59               10.2609 %
Kappa statistic                          0.6825
Mean absolute error                      0.1302
Root mean squared error                  0.264
Relative absolute error                 40.6189 %
Root relative squared error             65.9877 %
Coverage of cases (0.95 level)          99.8261 %
Mean rel. region size (0.95 level)      65.7391 %
Total Number of Instances              575

=== Confusion Matrix ===

   a   b   <-- classified as
 429  31 |   a = not
  28  87 |   b = variant
```

Listing 5-3. Confusion matrix for Naive Bayes

```
=== Summary ===

Correctly Classified Instances         571               99.3043 %
Incorrectly Classified Instances         4                0.6957 %
Kappa statistic                          0.978
Mean absolute error                      0.0089
Root mean squared error                  0.0663
Relative absolute error                  2.7682 %
Root relative squared error             16.565  %
Coverage of cases (0.95 level)          99.8261 %
Mean rel. region size (0.95 level)      50.5217 %
Total Number of Instances              575

=== Confusion Matrix ===

   a   b   <-- classified as
 460   0 |   a = not
   4 111 |   b = variant
```

Listing 5-4. Confusion matrix for multilayer perceptron

5.4.2 Attribute Evaluator

Character differences that do not indicate any spelling variation (e.g., [-a-]{+i+}, aawitan vs. aawitin) were pruned; word pairs with these character differences – totaling 67,059 – were removed. Visual inspection was also applied to ensure that no spelling variants were removed. The attribute evaluator was then used to rank which features are important. The attribute evaluator uses supervised learning to determine the “merit” of each feature (i.e., how likely a particular feature contributes to the classification). It provides as output the “merit” of the different features in descending order. The first in the list is always rank 1. Listing 5-5 shows the results. Wdiff or character difference has been ranked as the most important feature followed by the location of the character difference and the adjacent characters. These refer to a character trigram (i.e., the previous character, the character difference, and the succeeding character), indicating that it can be used to represent a possible spelling variant.

```
=== Attribute Selection on all input data ===

Search Method:
      Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 Attribute):
      Correlation Ranking Filter
Ranked attributes:
 0.1973   1 Wdiff
 0.1882   2 Wdiff Position
 0.1472   6 Char Right Type
 0.1464   3 Char Left
 0.0979   5 Char Right
 0.0799   7 Check Freq
 0.0799   8 Check Freq Inverted
 0.074    4 Char Left Type

Selected attributes: 1,2,6,3,5,7,8,4 : 8
```

Listing 5-5. Results for attribute selection

5.4.3 Character Differences Denoting a Spelling Variation

Character differences with more than three word pairs that have been annotated as spelling variants were selected. Together with visual inspection of character differences with no annotated spelling variants, a total of nine distinct character differences that denote a spelling variation were identified. The results are shown in Table 5-4. Sample word pairs and the trigram of the character differences are also shown. The second and fourth columns refer to the first variant while the third and fifth columns refer to the second variant. A sample feature set is shown in Table 5-5. A subset of these matches those reported in the literature (Ilao et al., 2011; Sentro ng Wikang Filipino – Diliman, 2008). Certain spelling changes however, have not been reported in any literature (e.g., [-o-]{+w+}, dinadalao vs. dinadalaw).

Table 5-4. Word pairs that denote spelling variations

Wdiff	**W1**	**W2**	**W1 Trigram**	**W2 Trigram**
[-c-]{+k+}	acalain	akalain	aca	aka
[-o-]{+u+}	abogadong	abugadong	bog	bug
[-d-]{+r+}	nadagdagan	naragdagan	ada	ara
[-e-]{+i+}	aatakehin	aatakihin	keh	kih
[-u-]{+w+}	aauitin	aawitin	aui	awi
[-l-]{+r+}	albularyo	arbularyo	alb	arb
[-i-]{+y+}	baitang	baytang	ait	ayt
[-o-]{+w+}	dinadalao	dinadalaw	ao_	aw_
[-b-]{+v+}	automobil	automovil	obi	ovi

Table 5-5. Sample feature set

Wdiff	Wdiff Position	Char Left	Type	Char Right	Type	Check Freq	Check Freq Inverted
[-c-]{+k+}	Middle	a	V	s	C	1	0
[-c-]{+k+}	Middle	i	V	u	V	0	1
[-c-]{+k+}	Middle	l	C	a	V	1	0
[-o-]{+u+}	Middle	n	C	m	C	0	1
[-o-]{+u+}	Middle	n	C	m	C	1	0
[-o-]{+u+}	Middle	a	V	t	C	1	0
[-d-]{+r+}	Middle	a	V	a	V	0	1
[-d-]{+r+}	Middle	a	V	a	V	0	1
[-d-]{+r+}	Middle	a	V	a	V	0	1
[-e-]{+i+}	Middle	d	C	t	C	1	0
[-e-]{+i+}	Middle	n	C	b	C	0	1
[-e-]{+i+}	Middle	g	C	s	C	0	1
[-u-]{+w+}	Middle	a	V	i	V	1	0
[-u-]{+w+}	Middle	a	V	a	V	1	0
[-u-]{+w+}	Middle	a	V	t	C	1	0
[-l-]{+r+}	Middle	u	V	o	V	0	1
[-l-]{+r+}	Middle	u	V	o	V	0	1
[-l-]{+r+}	Middle	a	V	a	V	1	0
[-i-]{+y+}	Middle	r	C	a	V	0	1
[-i-]{+y+}	Middle	s	C	o	V	1	0
[-o-]{+w+}	Last	a	V	_	_	1	0
[-o-]{+w+}	Last	a	V	_	_	1	0
[-b-]{+v+}	Middle	o	V	e	V	0	1
[-b-]{+v+}	Middle	e	V	e	V	1	0

5.4.4 Attributing Features

To determine the specific trigrams and other attributing features per character difference, the attribute selector was used on each set of word pairs. Listing 5-6, Listing 5-7, and Listing 5-8 show the results for [-c-]{+k+}, [-d-]{+r+}, and [-l-]{+r+}, respectively. It has been noted that the results for [-l-]{+r+} have low rank values. This may indicate that no definite set of attributing features were found. Also, the following wdiff have unary classes (i.e., all instances in the training data are variant pairs) and ranked attributes cannot be generated:

- [-o-]{+u+}
- [-e-]{+i+}
- [-i-]{+y+}
- [-u-]{+w+}
- [-o-]{+w+}
- [-b-]{+v+}

```
=== Attribute Selection on all input data ===

Search Method:
      Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 8 Attribute):
      Correlation Ranking Filter
Ranked attributes:
 1       1 Wdiff Position
 0.639   5 Char Right Type
 0.472   4 Char Right
 0.259   2 Char Left
 0.189   3 Char Left Type
 0.125   7 Check Freq Inverted
 0.125   6 Check Freq

Selected attributes: 1,5,4,2,3,7,6 : 7
```

Listing 5-6. Results for [-c-]{+k+} attribute selection

```
=== Attribute Selection on all input data ===

Search Method:
      Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 8 Attribute):
      Correlation Ranking Filter
Ranked attributes:
 1       5 Char Right Type
 1       1 Wdiff Position
 0.777   4 Char Right
 0.696   2 Char Left
 0.269   7 Check Freq Inverted
 0.269   6 Check Freq
 0.15    3 Char Left Type

Selected attributes: 5,1,4,2,7,6,3 : 7
```

Listing 5-7. Results for [-d-]{+r+} attribute selection

```
=== Attribute Selection on all input data ===

Search Method:
      Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 8 Attribute):
      Correlation Ranking Filter
Ranked attributes:
 0.429   5 Char Right Type
 0.27    7 Check Freq Inverted
 0.27    6 Check Freq
 0.258   4 Char Right
 0.228   3 Char Left Type
 0.186   2 Char Left
 0       1 Wdiff Position

Selected attributes: 5,7,6,4,3,2,1 : 7
```

Listing 5-8. Results for [-l-]{+r+} attribute selection

Together with manual analysis, a set of common features that attributes to a spelling variation were identified. These are shown in Table 5-6. However, no definite set of attributing features were determined for the following wdiff: [-o-]{+u+}, [-e-]{+i+}, and [-l-]{+r+}. This indicates that the character on the left and the character on the right almost vary per word pair and no definite set of character can be attributed to the character difference.

Table 5-6. Attributing features

Wdiff	Position	Char Left	Char Right
[-c-]{+k+}	Middle	Any	Any
[-o-]{+u+}	No definite set of features		
[-d-]{+r+}	Middle	Vowel	Vowel
[-e-]{+i+}	No definite set of features		
[-u-]{+w+}	Middle	Any	‘a’, ‘e’, ‘i’
[-l-]{+r+}	No definite set of features		
[-i-]{+y+}	Middle	Consonant	‘a’, ‘e’, ‘o’
[-o-]{+w+}	Last	‘a’	_
[-b-]{+v+}	Middle	Vowel	Vowel

By selecting word pairs with attributing features discussed in the previous table, a total of 396 out of 7,454 word pairs were identified as spelling variants. The breakdown is shown in Table 5-7. The wdiff [-c-]{+k+} has the highest number of spelling variants.

Table 5-7. Number of spelling variants per wdiff

Wdiff	Position
[-c-]{+k+}	180
[-d-]{+r+}	79
[-u-]{+w+}	68
[-i-]{+y+}	42
[-o-]{+w+}	20
[-b-]{+v+}	7

5.5 Variant Scoring

Variant scoring, the formula discussed in section 4.2.4, was applied next to determine which variant is a representative of the language model. Table 5-8 shows the results, which indicate that the second variant is more representative of the language model. Sample word pairs are shown in Table 5-9. However, there are exceptions, shown in Table 5-10. These are word pairs where the first variant has higher MWS than the second variant.

Table 5-8. Variant scoring results

Wdiff	**Higher**
[-c-]{+k+}	Variant 2
[-d-]{+r+}	Variant 2
[-u-]{+w+}	Variant 2
[-i-]{+y+}	Variant 2
[-o-]{+w+}	Variant 2
[-b-]{+v+}	Variant 2

Table 5-9. Sample word pairs and their weighted score

Wdiff	Word 1	W1 MWS	Word 2	W2 MWS
[-b-]{+v+}	debeloper	0.0116	developer	0.9884
[-b-]{+v+}	deliber	0.2708	deliver	0.7292
[-b-]{+v+}	gobernador	0.4951	governador	0.5049
[-c-]{+k+}	acsayahin	0.0598	aksayahin	0.4402
[-c-]{+k+}	alcalde	0.0083	alkalde	0.4917
[-c-]{+k+}	america	0.2749	amerika	0.7251
[-d-]{+r+}	dadaanan	0.3817	daraanan	0.6183
[-d-]{+r+}	dadagdagan	0.4649	daragdagan	0.5351
[-d-]{+r+}	dadamay	0.3916	daramay	0.6084
[-i-]{+y+}	estudianteng	0.0128	estudyanteng	0.4872
[-i-]{+y+}	estudio	0.2391	estudyo	0.2609
[-i-]{+y+}	gobierno	0.0018	gobyerno	0.4982
[-o-]{+w+}	dinadalao	0.1831	dinadalaw	0.8169
[-o-]{+w+}	dumadalao	0.1906	dumadalaw	0.8094
[-o-]{+w+}	dumalao	0.1714	dumalaw	0.8286
[-u-]{+w+}	aauitin	0.0833	aawitin	0.9167
[-u-]{+w+}	asauang	0.0572	asawang	0.9428
[-u-]{+w+}	bayauang	0.2500	bayawang	0.7500

Table 5-10. Exceptions

Wdiff	Word 1	W1 MWS	Word 2	W2 MWS	Higher
[-b-]{+v+}	automobil	0.4521	automovil	0.0479	Variant 1
[-c-]{+k+}	agricultural	0.2755	agrikultural	0.2245	Variant 1
[-c-]{+k+}	director	0.5953	direktor	0.4047	Variant 1
[-c-]{+k+}	electoral	0.6115	elektoral	0.3885	Variant 1
[-c-]{+k+}	masaclolo	0.3250	masaklolo	0.1750	Variant 1
[-c-]{+k+}	political	0.5390	politikal	0.4610	Variant 1
[-c-]{+k+}	protocol	0.4661	protokol	0.0339	Variant 1
[-c-]{+k+}	sectoral	0.5115	sektoral	0.4885	Variant 1
[-c-]{+k+}	tatalicdan	0.2879	tatalikdan	0.2121	Variant 1
[-d-]{+r+}	dinadaan	0.5348	dinaraan	0.4652	Variant 1
[-d-]{+r+}	madaanan	0.5152	maraanan	0.4848	Variant 1
[-d-]{+r+}	madagdagan	0.5379	maragdagan	0.4621	Variant 1
[-d-]{+r+}	madamay	0.5021	maramay	0.4979	Variant 1
[-d-]{+r+}	madumihan	0.3030	marumihan	0.1970	Variant 1
[-d-]{+r+}	magdadagdag	0.5189	magdaragdag	0.4811	Variant 1
[-d-]{+r+}	makadagdag	0.5182	makaragdag	0.4818	Variant 1
[-d-]{+r+}	nadagdag	0.5475	naragdag	0.4525	Variant 1
[-d-]{+r+}	nadagdagan	0.5426	naragdagan	0.4574	Variant 1
[-d-]{+r+}	nadamay	0.5502	naramay	0.4498	Variant 1
[-d-]{+r+}	nagdudugtong	0.2903	nagdurugtong	0.2097	Variant 1
[-i-]{+y+}	dialogo	0.3065	dyalogo	0.1935	Variant 1
[-i-]{+y+}	familia	0.5283	familya	0.4717	Variant 1
[-i-]{+y+}	glorieta	0.7586	gloryeta	0.2414	Variant 1
[-i-]{+y+}	historia	0.8861	historya	0.1139	Variant 1
[-i-]{+y+}	kolehio	0.4167	kolehyo	0.0833	Variant 1
[-i-]{+y+}	malaria	0.8209	malarya	0.1791	Variant 1
[-i-]{+y+}	material	0.6772	materyal	0.3228	Variant 1
[-i-]{+y+}	memoria	0.5242	memorya	0.4758	Variant 1
[-i-]{+y+}	monasterio	0.3800	monasteryo	0.1200	Variant 1
[-i-]{+y+}	pianista	0.3289	pyanista	0.1711	Variant 1
[-i-]{+y+}	plegaria	0.6455	plegarya	0.3545	Variant 1
[-i-]{+y+}	tenienteng	0.2708	tenyenteng	0.2292	Variant 1

5.6 Tagged Data

The entire tagged data (Manguilimotan and Matsumoto, 2011) contains a total of 93,321 entries. Table 5-11 shows the frequency count per tag. For this research, the scope is only on verbs in the actor focus because of the high number of verb affixes (Schachter and Otanes, 1972) and because of the exclusivity (i.e., they only take one form of affix given one aspect) reported in literature (Endriga, 2011). Out of the 7,546 words tagged as verbs, only 3,470 contain affixes. The rest are root words (e.g., “sabi”, “alam”, “maging”) and inflected words that were tagged as root words (e.g., “nagiging” was tagged as a root word).

Table 5-11. Frequency count per tag

Part-of-speech	Frequency
Conjunctions	16,113
Cardinal Marker	1,753
Determiner	8,779
Adjective	4,486
Lexical Marker	1,767
Noun	24,527
Pronoun	19,617
Adverb	7,354
Verbs	7,546
Others	1,379
Total	93,321

A total of 2,245 verbs (VB) in the actor focus, shown in Table 5-12, were used. The different aspects considered for this research are as follows:

- COAF – contemplated (e.g., magsasayaw);
- IMAF – imperfective (e.g. nag-aaral);
- INAF – infinitive (e.g., magtungo); and
- PFAF – perfective (e.g., nagtungo).

Table 5-12. Frequency count for verbs

Part-of-speech Tag	Frequency
VB-COAF	177
VB-IMAF	449
VB-INAF	783
VB-PFAF	836
Total	2,245

Emphasis is given on the following affixes: -um-, nag-, na-, and their counterparts in other aspects.

5.7 Data Association

The tokens together with the affixes were treated as transactions and Apriori, an algorithm for association rule mining, was applied. The top 11 is shown in Table 5-13. Only those with a confidence value of 1.0 were generated and those with lower confidence values were discarded. A confidence value of one is interpreted as the only affix used for that word (e.g. "dumating" instead of "nagdating") thus, exclusivity. The confidence value is defined as the frequency count of the affix and the lemma appearing together over the frequency count of the lemma.

Table 5-13. Top 11 based on frequency

No.	Lemma	Freq (Lemma)	Affix	Freq (Lemma and Affix)	Confidence
1	dating	44	_ um _ _	44	1
2	mula	9	nag _ _ _	9	1
3	tuto	7	na _ _ _	7	1
4	taglay	7	nag _ _ RDPL	7	1
5	hiling	5	_ um _ _	5	1
6	ubos	4	na _ _ _	4	1
7	aksaya	4	nag _ _ _	4	1
8	ako	4	nang _ _ _	4	1
9	akong	4	nang _ _ _	4	1
10	tugis	4	_ um _ RDPL	4	1
11	nais	4	nag _ _ RDPL	4	1

5.8 Cleaning

The presence of noise data can be noticed from the table. For instance, "nang-ako" should be "nangako". This prompted cleaning and pruning:

- manual inspection to remove noise data; and
- entries whose frequency count is below 3 were pruned.

The results for -um-, na-, and nag- are shown in Table 5-14, Table 5-15, and Table 5-16, respectively. This means that the words listed as taking the affix -um- do not take any other form of affix in that aspect and focus. However, there is no literature that supports the results for na- and nag-.

Table 5-14. Results for -um-

Lemma	Infinitive
dating	dumating
hiling	humiling
bagsak	bumagsak
sapi	sumapi
hinto	huminto
dalaw	dumalaw

Table 5-15. Results for na-

Lemma	Infinitive	Perfective	Imperfective	Contemplated
tuto	matuto	natuto	natututo	matututo
halal	mahalal	nahalal	nahahalal	mahahalal
mana	mamana	namana	namamana	mamamana
kulong	makulong	nakulong	nakukulong	makukulong
batid	mabatid	nabatid	nababatid	mababatid

Table 5-16. Results for nag-

Lemma	Infinitive	Perfective	Imperfective	Contemplated
mula	magmula	nagmula	nagmumula	magmumula
aksaya	mag-aksaya	nag-aksaya	nag-aaksaya	mag-aaksaya
pasya	magpasya	nagpasya	nagpapasya	magpapasya
tanong	magtanong	nagtanong	nagtatanong	magtatanong
hiwalay	maghiwalay	naghiwalay	naghihiwalay	maghihiwalay

5.9 Statistical Machine Translation

Approximately a total of 20 annotated documents were encoded and the corpus contains 100 lines. However, a third of the sentences do not contain any correction. These sentences refer to the usage of the lexical marker "ay". These were removed and the parallel corpus was cleaned using the following Moses scripts:

- tokenizer.perl – inserts spaces between words and punctuations; and
- clear-corpus-n.perl – very long sentences, and empty sentences.

The resulting parallel corpus contains a total of 62 lines. The phrase table rules were then generated using the default setting with up to a trigram language model. A sample is shown in Table 5-17. The phrase table scores refer to the following:

- inverse phrase translation probability f(f|e)
- inverse lexical weighting lex(f|e)
- direct phrase translation probability f(e|f)
- direct lexical weighting lex(e|f)

Table 5-17. Sample phrase table rules

Marked	Correction	Phrase Table Scores	Alignment
kwarter faynals	quarter finals	1 1 1 1	0-0 1-1
kwarter	quarter	1 1 1 1	0-0
labingisang	labing-isang	1 1 1 1	0-0

Together with manual analysis, phrase pairs that denote mistakes were identified. The entire list is shown in Table 5-18.

Table 5-18. Complete list of phrase pairs

Marked	Correction
computer	kompyuter
meroon	mayroon
binabalas	binabalasa
faynals	finals
ganun	ganon
green-houses	greenhouses
i-aalok	aalukin
kwarter	quarter
labingisang	labing-isang
meron	mayroong
meroong	may
meroong	mayroong
nagging	naging
pagaaral	pag-aaral
pagdedeal	pagdi-deal
pagkaibhan	pagkakaiba
palatuntunin	alituntunin
rekord	record
spesipikong	ispesipikong
standard	istandard
tradisiyonal	tradisyonal
Sa katapusan	Nang lumaon

5.10 Rule Development

For spelling variants, regular expressions (regex) were used to declare the pattern and regex replace was used for the suggestion. Variants that have a different variant scoring were declared as exceptions and the annotated data were used as examples. Listing 5-9 show an example rule. The pattern reflects the attributing values reported in Table 5-6 and the exception reflects those reported in Table 5-10.

```
<pattern case_sensitive="yes" mark_from="0">
        <token regexp="yes">[a-z]*c.*
        <exception regexp="yes">
                agricultural|director|electoral|political|protocol|sectoral
        </exception></token>
</pattern>
<message> Do you mean
<suggestion>
<match no="1" case_conversion="startlower" regexp_match="(.*)c(.*)" regexp_replace="$1k$2"/>
</suggestion></message>
<example correction="akalain" type="incorrect"><marker>acalain</marker>.</example>
<example type="correct"><marker>akalain</marker></example>
```

Listing 5-9. Sample rule for spelling variations

For affix usage, the words were declared in the pattern together with the affix mag- and nag-. Regex replace was used for the suggestion, transforming the token into its –um- form. The data available was used as examples. Listing 5-10 show an example rule. The words in Table 5-14 are declared in the pattern.

```
<pattern case_sensitive="yes">
        <token regexp="yes">[mn]ag(dating|dayo|hiling|bagsak|sapi|hinto|dalaw)</token>
</pattern>
<message>Do you mean
<suggestion><match no="1" regexp_match="nag(.)(.*)" regexp_replace="$1um$2">
</match></suggestion>?</message>
<example correction="dumating" type="incorrect"><marker>nagdating</marker></example>
<example type="correct"><marker>dumating</marker></example>
```

Listing 5-10. Sample rule for affix usage

For common mistakes, the words/phrases were declared in the pattern and the correct form was declared in the suggestion. The entries in the phrase table rules were declared as examples. Listing 5-11 show an example rule. The words declared in the first column of Table 5-18 are declared in the pattern while the words declared in the second column are declared in the suggestion.

```
<pattern case_sensitive="no" mark_from="0">
        <token>computer</token>
</pattern>
<message>Do you mean <suggestion>kompyuter</suggestion>?</message>
<short>Common Mistakes</short>
<example correction="kompyuter" type="incorrect"><marker>computer</marker></example>
<example type="correct"> <marker>kompyuter</marker></example>
```

Listing 5-11. Sample rule for common mistakes

In total, 6 new rules for spelling variations, 1 new rule for affix usage, and 22 new rules for common mistakes were generated. Except for affix usage, all rules were generated using a template. The LanguageTool community would benefit from the statistics based rule generation framework through faster rule generation and wider error checking coverage.

5.11 Comparison with Existing Literature

Several variations have been reported. Table 5-19 shows a table of comparison between the results of this research and those reported in literature. This research covered more single-letter spelling variations than any literature. Two character differences were not reported in this research due to low number of instances. These are [-s-]{+z+} and [-f-]{+p+}. The word pairs are shown in Table 5-20 and Table 5-21, respectively. It can be noticed for [-s-]{+z+} character difference that variant 1 is more dominant in terms of frequency. For [-f-]{+p+} character difference, the variant 2 is more dominant in terms of frequency. Also, [-u-]{+w+}, [-l-]{+r+}, and [-o-]{+w+} spelling variations in Tagalog were not covered in literature. However, one study (Ilao, Santos, and Guevara, 2012) identified [-o-]{+w+} as a spelling variation in Cebuano/Visayan. Linguistic phenomena involving more than one letter replacement or insertion were not covered in this research: assimilation (e.g., sangla vs. sanla), syncope (e.g., dakipin vs. dakpin), affix reduplication (e.g., makakaunawa vs. makauunawa), code-switching (e.g., nagpi-friendster vs. nagfrie-friendster) and other spelling variations (e.g., puwede vs. pwede).

Table 5-19. A comparison of different spelling variations in literature

Results	(Ilao et al., 2011)	(Zuraw, 2006)
[-c-]{+k+}	[-c-]{+k+}	N/A
[-o-]{+u+}	[-o-]{+u+}	[-o-]{+u+}
[-d-]{+r+}	N/A	[-d-]{+r+}
[-e-]{+i+}	[-e-]{+i+}	[-e-]{+i+}
[-u-]{+w+}	N/A	N/A
[-l-]{+r+}	N/A	N/A
[-i-]{+y+}	[-i-]{+y+}	N/A
[-o-]{+w+}	N/A	N/A
[-b-]{+v+}	[-b-]{+v+}	N/A
N/A (5 instances)	[-s-]{+z+}	N/A
N/A (7 instances)	[-f-]{+p+}	N/A

Table 5-20. Word pairs with [-s-]{+z+} character difference

Word 1	Frequency	Word 2	Frequency
arsobispo	109	arzobispo	22
magasin	345	magazin	16
mansanas	297	manzanas	30
mestiso	110	mestizo	21
postiso	31	postizo	21

Table 5-21. Word pairs with [-p-]{+f+} character difference

Word 1	Frequency	Word 2	Frequency
definisyon	12	depinisyon	89
kafatid	86	kapatid	12,005
profeta	36	propeta	487
referendum	27	reperendum	47
referensiya	18	reperensiya	31
reforma	25	reporma	983
transformasyon	26	transpormasyon	77

A comparison with existing style is shown in Table 5-22 and in Table 5-23. The words were taken from the examples in the guides. The proposed variant scoring matches the style proposed by Sentro ng Wikang Filipino (SWF) with 100% precision, 30% recall, and 30% accuracy, and matches the style proposed by the Komisyon sa Wikang (KWF) Filipino with 100% precision, 60% recall, and 60% accuracy. This is an indication that the style proposed by KWF is more inclined with the variant scoring. One variant pair is found in both styles: “politika” vs. “pulitika”. SWF deems “politika” as correct while KWF deems “pulitika” is correct. A close look at the style proposed by KWF (Komisyon sa Wikang Filipino), reveals the use of “eskandalo”, “espesyal”, and “estilo” to signify that the words originated from Spanish.

Table 5-22. Writing style proposed by SWF (Sentro ng Wikang Filipino – Diliman, 2008)

Correct	Frequency	Incorrect	Frequency	Higher MWS
ano-ano	546	anu-ano	1,282	Variant 2
sino-sino	91	sinu-sino	490	Variant 2
halo-halo	48	halu-halo	44	Variant 2
salo-salo	42	salu-salo	154	Variant 2
estilo	1,127	istilo	316	Variant 2
estasyon	117	istasyon	864	Variant 2
estudyante	3,173	istudyante	56	Variant 1
estadistika	65	istadistika	26	Variant 2
espiritu	757	ispiritu	61	Variant 1
espesyal	1,095	ispesyal	27	Variant 1
estrikto	0	istrikto	92	Variant 2
eskandalo	346	iskandalo	586	Variant 2
politika	627	pulitika	5,547	Variant 2
opisina	2,132	upisina	65	Variant 1
kombinasyon	99	kumbinasyon	186	Variant 2
tradisyonal	448	tradisyunal	356	Variant 1
kompleto	34	kumpleto	633	Variant 2
kompanya	1,867	kumpanya	4,447	Variant 2
kontrata	2,332	kuntrata	0	Variant 1
komersiyal	97	kumersiyal	0	Variant 2

Table 5-23. Writing style proposed by KWF(Komisyon sa Wikang Filipino, 2013)

Correct	Frequency	Incorrect	Frequency	Higher MWS
iskandalo	586	eskandalo	346	Variant 1
istasyon	864	estasyon	117	Variant 1
istilo	316	estilo	1,127	Variant 1
minudo	0	menudo	19	Variant 2
nigatibo	0	negatibo	282	Variant 2
kuryente	2,155	koryente	207	Variant 2
dunasyon	0	donasyon	496	Variant 2
kumpanya	4,447	kompanya	1,867	Variant 1
sumbrero	144	sombrero	155	Variant 1
pulitika	5,547	politika	627	Variant 1

The results of the data association were also compared with an existing literature (Endriga, 2011). Table 5-24, Table 5-25, and Table 5-26 show a comparison:

- Four out of six words that take the –um- affix to focus the actor were also reported in the literature while one was reported to focus the theme;
- Two out of five words that take the na- affix to focus the theme were reported in the literature; and
- Three out of five words that take the nag- affix were reported in the literature.

Table 5-24. A comparison of the –um- affix usage

Results	(Endriga, 2011)
bagsak	bagsak (theme e.g., bumagsak)
dalaw	dalaw
dating	dating
hiling	hiling (non-volitional agent)
hinto	hinto (non-volitional agent)
sapi	N/A

Table 5-25. A comparison of the na- affix usage

Lemma	(Endriga, 2011)
tuto	N/A
halal	halal (theme e.g., "nahalal")
mana	N/A
kulong	kulong (theme e.g., "nakulong")
batid	N/A

Table 5-26. A comparison of the nag- affix usage

Lemma	(Endriga, 2011)
mula	mula
aksaya	N/A
pasya	N/A
tanong	tanong
hiwalay	hiwalay (dual-reciprocal)

6 Conclusion

A statistics-based rule generation framework for Filipino style and grammar checking has been presented in this paper. Monolingual corpora, annotated documents, as well as a tagged data were collected. The monolingual corpus was modeled and machine learning was used to aid in detecting spelling variations. A scoring mechanism was proposed to determine which variant represents the language model more. The tagged data was processed and data association was applied to determine affix usage. Lastly, a subset of the annotated documents was digitized and used as training data for a statistical machine translation engine to determine common mistakes made. A total of 396 variant pairs, 16 affix usage, and 22 phrase pairs were generated and transformed into rules. For spelling variations, the spelling with the lower score is declared in the pattern while the spelling with the higher score is declared in the suggestion; for affix usage, other forms of affixes were declared in the pattern while the affix with high confidence value was declared in the suggestion; and lastly for common mistakes, the token was declared in the pattern while the correction is declared in the suggestion. A subset of these linguistic phenomena was reported in the literature, an indication that the framework can be used to automate linguistic tasks. The proposed variant scoring matches the style proposed by Sentro ng Wikang Filipino (SWF) with 30% recall and matches the style proposed by the Komisyon sa Wikang (KWF) Filipino with 60% recall, an indication that the style proposed by KWF is more inclined with the variant scoring. Also, with the use of the framework, certain linguistic phenomena not covered by existing literature were generated such as the u vs. w spelling change (e.g., aauitan vs. aawitan). This highlights the potential corpus-based analyses have in language policy and planning. As future work, a policy paper could be drafted in coordination with experts in language planning. Additionally, SMT can be used with more data and a stemmer be developed as aid in producing the tagged data.

References

Alam, M. J., Naushad UzZaman and Mumit Khan. (2006). N-gram Based Statistical Grammar Checker for Bangla and English Pages. Proceedings of the 9th International Conference on Computer and Information Technology.

Allman, T., Stephen Beale, and Richard Denton. (2014). Toward an Optimal Multilingual Natural Language Generator: Deep Source Analysis and Shallow Target Analysis. Proceedings of the 10th National Natural Language Processing Research Symposium.

Ang, M., Sonny G. Cagalingan, Paulo Justin U. Tan, and Reagan C. Tan. (2002). FiSSAn: Filipino Sentence Syntax and Semantic Analyzer. Undergraduate Thesis. De La Salle University, Manila.

Atzmueller. M. (2012). Data Mining. Chap. 5 in Applied Natural Language Processing: Identification, Investigation and Resolution. Hershey, PA: IGI Global.

Bautista, M.L.S., Loy V. Lising, and Danilo T. Dayag. (2004). ICE-Philippines Lexical Corpus - CD-ROM. London: International Corpus of English.

Bergen, B. (2001). Nativization Processes in L1 Esperanto. Journal of Child Language, 28: 575-595.

Castilo, M.A., Matthew Phillip Go, Alron Jan Lam, Oliver Brian Syson, Peigen Xu, Ethel Ong, and Stephen Beale. (2014). Building a Simple Linguist's Assistant for Tagalog. Proceedings of the 10th National Natural Language Processing Research Symposium.

Cavnar, W. and John M. Trenkle. (1994). N-Gram-Based Text Categorization. Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval.

Cunningham, H., Diana Maynard, Kalina Bontcheva, Valentin Tablan. (2002). GATE: an Architecture for Development of Robust HLT Applications. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.

Dempster, A., Nan Laird, and Donald Rubin. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, 39(1): 1–38.

Dice, L.R. (1945). Measures of the Amount of Ecologic Association between Species. Ecology, 26(3): 297-302.

Dimalen, D.M. and Editha D. Dimalen. (2007). An OpenOffice Spelling and Grammar Checker Add-in Using an Open Source External Engine as Resource Manager and Parser. Proceedings of the 4th National Natural Language Processing Research Symposium.

Dimalen, D.M. and Rachel Edita O. Roxas. (2007). AutoCor: A Query Based Automatic Acquisition of Corpora of Closely-related Languages. Proceedings of the 21st Pacific Asia Conference on Language, Information, and Computation.

Dita, S., Rachel Edita O. Roxas, and Paul Inventado. (2009). Building online corpora of Philippine languages. Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation.

Endriga, D.A. (2011). Refining the Agent. Proceedings of the 11th Philippine Linguistics Congress.

Feldman, A. and Jirka Hana. (2010). A positional tagset for Russian. In: Proceedings of the 7th International Conference on Language Resources and Evaluation, European Language Resources Association.

Ilao, J., Rowena Cristina L. Guevara, Virgilio D. Llenaresas, Eilene Antoinette G. Narvaez, Jovy M. Peregrino. (2011). Bantay-Wika: towards a better understanding of the dynamics of Filipino culture and linguistic change. Proceedings of the 9th Workshop on Asian Language Resources Collocated with IJCNLP 2011.

Ilao, J., Timothy Israel D. Santos, and Rowena Cristina L. Guevara. (2012). *Komparatibong Analisis ng Aktuwal na Gamit ng Wika at mga Piling Pamantayan sa Gramatika at Ortograpiya sa Filipino, Sebwano-Bisaya at Ilokano: Lapit Batay sa Korpus* / Comparative analysis of actual language usage and selected grammar and orthographical rules for Filipino, Cebuano-Visayan and Ilokano: a Corpus-based Approach. Daluyan: Journal ng Wikang Filipino,

Klein, D. and Chris Manning. (2003). Accurate unlexicalized parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics.

Komisyon sa Wikang Filipino. (2013). Binagong Gabay sa Ortograpiya ng Wikang Filipino.

Konchady, M. (2009). Detecting Grammatical Errors in Text using a Ngram-based Ruleset. Retrieved from: http://emustru.sourceforge.net/detecting_grammatical_errors.pdf

Kondrak, G. (2005). N-gram similarity and distance. Proceedings of the 12th International Conference on String Processing and Information Retrieval.

Jasa, M., Justin O. Palisoc, and Martee M. Villa. (2007). Panuring Pampanitikan (PanPam): A Sentence Syntax and Semantic Based Grammar Checker for Filipino. Undergraduate Thesis. De La Salle University, Manila.

Lam, A.J., Ivan Paner, Jules Matthew Macatangay, and Duke Danielle Delos Santos. (2014). Classifying Typhoon Related Tweets. Proceedings of the 10th National Natural Language Processing Research Symposium.

Levenshtein, V. (1965). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Doklady Academii Nauk SSSR, 163(4): 845-848.

Lynch, J. (2008). The English Language: A User's Guide. Newburyport, MA: Focus Publishing/R. Pullins Company.

Manguilimotan, E. and Yuji Matsumoto. (2011). Dependency-based Analysis for Tagalog Sentences. Proceedings of the 25th Pacific Asia Conference on Language, Information, and Computation.

Miguel, D. and Rachel Edita Roxas. (2007). Comparative analysis of Tagalog part-of-speech (POS) taggers. Proceedings of the 4th National Natural Language Processing Research Symposium.

Naber, D. (2003). A Rule-Based Style and Grammar Checker. Diploma Thesis. Bielefeld University, Bielefeld.

Oco, N. and Allan Borra. (2011). A Grammar Checker for Tagalog using LanguageTool. Proceedings of the 9th Workshop on Asian Language Resources Collocated with IJCNLP 2011.

Oco, N. and Rachel Edita Roxas. (2012). Pattern Matching Refinements to Dictionary-based Code-Switching Point Detection. Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation.

Oco, N., Jason Wong, Joel Ilao, and Rachel Edita Roxas. (2013a). Code-Switches using Word Bigram Frequency Count. Proceedings of the 9th National Natural Language Processing Research Symposium.

Oco, N., Joel Ilao, Rachel Edita Roxas, Leif Romeritch Syliongka. (2013b). Measuring Language Similarity using Trigrams: Limitations of Language Identification. Proceedings of the 3rd International Conference on Recent Trends in Information Technology.

Oco, N., Leif Romeritch Syliongka, Joel Ilao, Rachel Edita Roxas. (2013c). Dice's Coefficient on Trigram Profiles as Metric for Language Similarity. Proceedings of the 16th Oriental COCOSDA.

Oco, N., Leif Romeritch Syliongka, Joel Ilao, and Rachel Edita Roxas. (2014a). N-gram based Language Identification and Rule-based Grammar Checking. Proceedings of the 14th Philippines Computing Science Congress.

Oco, N., Raquel Sison-Buban, Leif Romeritch Syliongka, Rachel Edita Roxas, and Joel Ilao. (2014b). Ang Paggamit ng Trigram Ranking Bilang Panukat sa Pagkakahalintulad at Pagkakapangkat ng mga Wika/Trigram Ranking: Metric for Language Similarity and Clustering. Malay, 26 (2), pp: 53-68.

Rabo, V. (2004). TPOST: A template-based, n-gram part-of-speech tagger for Tagalog. Graduate Thesis. De La Salle University, Manila.

Rabo, V. and Charibeth K. Cheng. (2006). TPOST: A Template-based Part-of-Speech Tagger for Tagalog. Journal of Research in Science, Computing, and Engineering, 3(1).

Ramos, T. (1971). Makabagong Balarila ng Pilipino. Manila: Rex Bookstore.

Roxas, R.E., Charibeth K. Cheng, and Nathalie Rose T. Lim. (2009). Philippine Language Resources: Trends and Directions. Proceedings of the 7th Workshop on Asian Langauge Resource.

Roxas, R.E., Allan Borra, Charibeth Cheng, Nathalie Rose Lim, Ethel Ong, Michelle Wendy Tan. (2008). Building Language Resources for a Multi-Engine English-Filipino Machine Translation System. Language Resources and Evaluation, 42(2): 183-195.

Santos, L. (1939). Balarila ng Wikang Pambansa. Manila: Institute of Philippine Language.

Sentro ng Wikang Filipino – Diliman. (2008). Gabay sa Editing sa Wikang Filipino. Quezon City: Unibersidad ng Pilipinas.

Schachter, P. and Fe T. Otanes. (1972). Tagalog Reference Grammar. Berkeley, CA: University of California Press.

Schumaker, R. and Hsinchun Chen. (2010). Interaction Analysis of the ALICE Chatterbot: A Two-Study Investigation of Dialog and Domain Questioning. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 40(1): 40-51.

Syliongka, L.R. and Nathaniel Oco. (2014). Using Language Modeling and Data Association to Perform Named Entity Recognition. Proceedings of the 10th National Natural Language Processing Research Symposium.

Wang, T., Yaoyong Li, Kalina Bontcheva, Hamish Cunningham, and Ji Wang. (2006). Automatic Extraction of Hierarchical Relations from Text. Proceedings of the 3rd European Conference on the Semantic Web: Research and Applications.

Yeong, Y.-L. and Tien-Ping Tan. (2010). Language Identification of Code-Switching Malay-English Words Using Syllable Structure Information. Proceedings of the 2nd Workshop on Spoken Languages Technologies for Under-Resourced Languages.

Zuraw, K. (2006). Using the Web as a Phonological Corpus: a Case Study from Tagalog. Proceedings of the 2nd International Workshop on Web as Corpus.

Appendix A. Email Communication

This section shows email communications cited in this research.

With Mr. Mark Johnson (February 01, 2011)

Professor of Language Sciences (CORE)
Director, Centre for Language Sciences (CLaS)
Department of Computing
Faculty of Science
Macquarie University
Sydney, Australia
mark.johnson@mq.edu.au

Hi Nathaniel,

I think most grammar-checkers are simple regular-expression pattern matchers -- they look for a relatively small set of possible errors. The idea is that it's not enough to detect an error; the program should also propose a correction. This is not straight-forward with a probabilistic grammar; it's hard to tell exactly where the error is, and how it should be fixed.

However, I've heard that the grammar checker in Microsoft Word (TM) does use a probabilistic model, so I guess it can be made to work. Microsoft isn't saying exactly how it works, though!

Best,
Mark Johnson

With Mr. Manu Konchady (May 08, 2011)
Mustru Search Services
mkonchady@yahoo.com

1. In your paper, you mentioned that one of the drawbacks of LanguageTool (http://www.languagetool.org/) is its low recall rate because the "number of rules to cover a majority of the grammatical errors is much larger". Could you please elaborate this statement.

Well, I believe that the grammar rules in LanguageTool must be manually generated. Unfortunately, language can be used in almost infinite ways and it takes a large number of rules to spot all possible errors. One example of a rule from LanguageTool is shown in the paper. You can imagine the effort required to generate several thousand such rules.

2. What is the relationship between the amount of rules and the recall rate of a Manual Rule-Based system?

Since, the number of rules in a manual system is less than the number of rules in an automated system, a grammar checker using the manual rules will miss many of the possible errors leading to lower recall.

3. What are the advantages of a Manual Rule-Based system over an Automatic Rule-Based system?

An automatic system will create rules based on statistics in a tagged corpus. However, the tagged corpus may not cover all possible instances of tag patterns and therefore, the automatic rules may not generate all possible language POS tag patterns.

It is easy to add rules to manual system to gradually build a more precise grammar checker over time. Any errors detected by the automatic system are almost certain to be errors. Therefore precision is high.

Regards,

Manu Konchady

With Mr. Davis Dimalen (2011)
De La Salle University Alumni
d_dimalen@yahoo.com

The problems observed from the corpus extracted from ICE-PH. The problems that were considered in this report are code switching points that were tagged by the "<indig></indig>" tag pairs. The items below are the descriptions of the different problems discovered in the corpus. The problematic elements where marked red.

1. Some tags were not closed
Example:
These things <indig> naka- boutique
Oh she she she tried it for the
soap but it 's kinda big
She make it she 'll make it smaller
<indig> para </indig> it 's easier to repack it

2. Improper use of close </indig> tag.
Example:
I was so <indig> lugi pala <indig> the other day
Why
Yesterday I missed all my classes you know
that

3. Some words were tagged as indigenous when they are not really indigenous
Example:
Especially <indig> yung mga mga ano yung mga trying
times </indig>

The above problems did not only occur once in the corpus but several times.

Appendix B. Word Unigram

hindi 231660
isang 198385
kung 166616
niya 142099
para 141337
siya 139178
nang 139067
naman 129414
lang 125746
kanyang 121829
dahil 112044
lamang 82785
nila 72313
sila 71784
kanilang 70435
kaya 65765
lahat 65050
walang 58938
nito 57172
upang 56721
ngayon 53802
niyang 48410
siyang 47890
wala 47411
mula 45978
pang 44606
aking 43460
dapat 41486
natin 40235
noong 39468
kanya 38996
pero 38978
ating 38268
buhay 37376
bilang 37274
iyong 37192
namin 36236
dito 36083
kong 35302
anak 34689
kahit 34048
kami 33045
ibang 32845
loob 32570
araw 32542
akin 31354
bansa 30570
nasa 30133
yung 29078
maging 29043
saan 28298
dalawang 27769
kasi 27620
ilang 27307
tayo 26708
bahay 26281
kayo 25919
hanggang 25359
akong 25315
bagong 24952
taong 24885
bago 24380
taon 24147
mong 24010
itong 23711
naging 23701
alam 23661
kasama 23583
habang 23137
talaga 23039
panahon 22570
laban 22570
sina 22118
bagay 22009
tungkol 21636
pamamagitan 21525
kapag 21088
sinabi 20998
kanila 20960
unang 20785
nilang 20782
lalo 20588
maraming 20010
iyon 19984
umano 19923
silang 19824
dating 19675
bayan 19394
bakit 19362
matapos 19168
sabi 19075
gusto 18991
sarili 18987
namang 18770
kundi 18556
ninyo 18200
noon 18043
parang 17994
inyong 17993
ngunit 17974
doon 17889
marami 17850
babae 17767
nina 17572
buong 17568
sana 16835
nitong 16743
ring 16743
maaaring 16447
tulad 16446
tunay 16377
pala 16026
oras 15892
biktima 15814
kailangan 15781
asawa 15696
pamilya 15249
puso 15181
sabihin 14798
bata 14776
huwag 14770
malaking 14714
aming 14446
nating 14420
gabi 14372
dalawa 14287
halos 14226
lugar 14104
trabaho 14055
ikaw 13876
kaniyang 13820
nasabing 13743
agad 13656
lalaki 13639
muna 13588
ginawa 13515
sino 13513
bawat 13270
problema 13096
kahapon 13014
kaibigan 12954
ibig 12644
tatlong 12619
rito 12575
kang 12541
baka 12528
ayon 12432
gobyerno 12430
rights 12369
dahilan 12278
gaya 12275
ayaw 12129
iyan 12072
mang 12020
kapatid 12005
pagkatapos 11745
buwan 11607
magiging 11600
nangyari 11595
higit 11593
patuloy 11409
gawin 11343
katawan 11293
reserved 11267
kaso 11202
gayon 11193
lalong 11188
pangalan 11051
amin 10971
bahagi 10941
pera 10879
tanong 10774
muli 10754
mundo 10705
dumating 10676
talagang 10475
sagot 10456
batas 10421
tila 10414
paano 10377
pamahalaan 10368
ngayong 10344
mahal 10256
totoo 10244
nakita 9966
paraan 9931
ginagawa 9912
kamay 9880
that 9880
bang 9710
lupa 9693
huling 9648
magandang 9593
nung 9574
magulang 9545
inyo 9545

saka 9516	sakit 8028	kabilang 7252	dibdib 6240
pati 9442	pag-ibig 8012	gayong 7092	dati 6220
sariling 9407	apat 7999	linggo 7086	tapos 6209
show 9398	opisyal 7917	dalaga 7018	kandidato 6183
ganitong 9317	isip 7907	yaon 6993	isyu 6172
mata 9220	ganito 7893	ding 6929	tuwing 6166
tubig 9133	gagawin 7858	time 6872	batang 6136
tayong 9062	naturang 7824	siguro 6869	galit 6112
mukha 8989	malapit 7705	kapwa 6791	bukas 6109
suspek 8900	maganda 7691	ilalim 6790	matagal 6076
kaming 8867	salita 7649	mayroon 6787	halaga 6048
kita 8866	galing 7641	minsan 6748	pelikula 6042
magkaroon 8860	tulong 7641	presyo 6745	nakikita 6017
atin 8834	anumang 7633	harap 6705	laro 6015
sinasabi 8814	grupo 7629	hirap 6683	sobrang 5989
naming 8789	malaki 7627	malaman 6591	yaong 5983
pagiging 8727	anong 7532	binata 6494	madalas 5983
pahayag 8719	pagkakataon 7501	gumawa 6451	tingin 5969
susunod 8659	kailangang 7487	mataas 6385	kaniya 5962
wika 8571	tuloy 7479	balita 6374	labas 5952
nakaraang 8562	nguni 7453	kababayan 6372	mismo 5943
mabuti 8508	maaari 7451	malakas 6362	miyembro 5889
makita 8300	mahirap 7329	gustong 6314	maliit 5888
nais 8281	with 7309	huli 6290	nagkaroon 5885
muling 8150	tama 7264	pulis 6267	pagdating 5869
kayong 8067	umaga 7254	pagkain 6265	ulit 5776

Appendix C. Personal Vitae

Nathaniel Oco
Researcher
Center for Language Technologies
College of Computer Studies
De La Salle University-Manila
nathan.oco@delasalle.ph
+639178477549