Gramatika: A Grammar Checker for the Low-Resourced Filipino Language

Matthew Phillip Go, Nicco Nocon and Allan Borra
De la Salle University
Manila, Philippines
{matthew_phillip_go, nicco_louis_nocon, allan.borra}@dlsu.edu.ph

***Abstract*— This research focuses on the implementation of Gramatika, a grammar checker designed for the Filipino language given its available resources and linguistic tools. The checker uses hybrid n-grams generated from n-grams of words, part-of-speech tags, and lemmas of grammatically-correct texts. It covers a variety of error types including those unique in Filipino: wrong word form, and incorrectly merged / unmerged words. The grammar checker performed 64% accuracy on producing the correct suggestions on erroneous phrases and 85% on error-free texts when using Part-of-Speech (POS) tags from a Hybrid POS tagger (HPOST) for Filipino. Recommendations to improve Gramatika is to implement linguistic tools such as constituency parser, incorrect affix detection system, and a spell checker for the Filipino language.**

***Index Terms*— Grammar Checking, Natural Language Processing, Language Analysis, Part-of-Speech Tagging, Filipino Language**

I. Introduction

Grammar checkers integrated inside word processing software such as Microsoft Word and Google Docs has provided writers of the English (and other covered) languages the convenience of automatic grammatical or spelling correction. These grammar checkers are able to identify run-on sentences, misspellings, disagreements between subjects and verbs, and other error types that need to be corrected. Such capabilities of error detection and correction clearly showcase the high level of research and development towards the English grammar. Corpora containing grammatical errors made by English as Second Language (ESL) learners such as the NUCLE corpus, Cambridge Learner Corpus, large corpora of grammatically-correct texts (Web1T), ontology between English words have been significant in the development of English grammar checkers. Linguistic tools such as shallow/ constituency parsers, treebanks, and high accuracy POS taggers are also important tools to process texts for these English grammar checkers [8], [10].

On the other hand, Filipino resources and tools are far behind their English counterparts. Size-wise, Filipino resources are small relative to English [3] and does not have a corpora containing grammatical errors similar to Cambridge Learner Corpus. In terms of linguistic tools, Filipino does not have any usable shallow / constituency parsers and no treebanks designed yet. At least Filipino has a decent Part-of-Speech (POS) tagger such as [9]'s successor Hybrid POS Tagger (HPOST), developed as part of this research, which achieves around 91% tagging accuracy, slightly lower than the Stanford POS tagger for English that achieves at least 97% accuracy [6]. Additionally, stemmers and morphological analyzers, with their own limitations, are also available for the Filipino language to handle its complex high-morphology linguistic phenomena.

Bounded with these limitations in Filipino, it was observed that there are promising researches in English made use of grammatically-correct texts and infers grammar rules from them to be used for error detection and correction [4], [13], [5], [7], [1]. Specifically, EdIt [4] and Lexbar [13] has closely similar ways of inferring grammar rules from texts. These works make use of words, POS tags, and lemmas, and converts them to rules. EdIt uses a collocations-based approach which creates rules around collocating words (ex. play & role) generating the rule 'play $\sim$ role *in* V-ing' from the sentence '*He played an important role in closing this deal*'. This rule can be then used to correct an erroneous sequence such as '*She plays a significant role in finish this project*' to suggest to correct the word *finish* as *finishing*. In terms of error types detected and corrected, Lexbar only tackles errors correctable using substitution while EdIt covers errors correctable by substitution, insertion, and deletion with minor prioritization for substitution-correctable errors wherein the correction and erroneous word belong to the same word group.

Gramatika 2.0 utilizes available resources in Filipino: POS tagger, morphological analyzers, and grammatically-correct texts to infer grammar rules (called *hybrid n-grams*) and targets more diverse error types found in Filipino writing correctable by: substitution (affixation errors, misspellings, wrong word usage but same word group, wrong word usage - different word group, merging of words, unmerging of a word, insertion, and deletion. Merging and unmerging corrections are for unique errors in the Filipino language which will be discussed in the later sections. Gramatika 2.0 (simply Gramatika in the later paragraphs) is an improved version of the initial implementation of the hybrid n-gram based grammar checker for Filipino which only used fewer hand-tagged POS tags [3]. The improved version uses system-produced POS tags which provided more training and testing data allowing for more comprehensive analysis.

978-1-5090-1134-6/17/$31.00 ©2017 IEEE

II. Overview of Gramatika

Gramatika[1] infers grammar rules from error-free POS-tagged and lemma-tagged Filipino texts based on a recent work [3]. It converts the words, POS tags, and lemmas into hybrid n-grams of sizes 2 to 7 that will be later used for error detection and correction. Two sets of POS tags and 2 sets of lemmas are used in this research: manually labeled by a linguist and produced by a POS tagger and lemmatizer tool, respectively. The aim of this research is to create an unsupervised grammar checker which automatically populates its grammar rules using texts in the web tagged using a POS tagger and a lemmatizer. To illustrated this, shown in Fig. 1 is the general view of Gramatika's resource and processes.

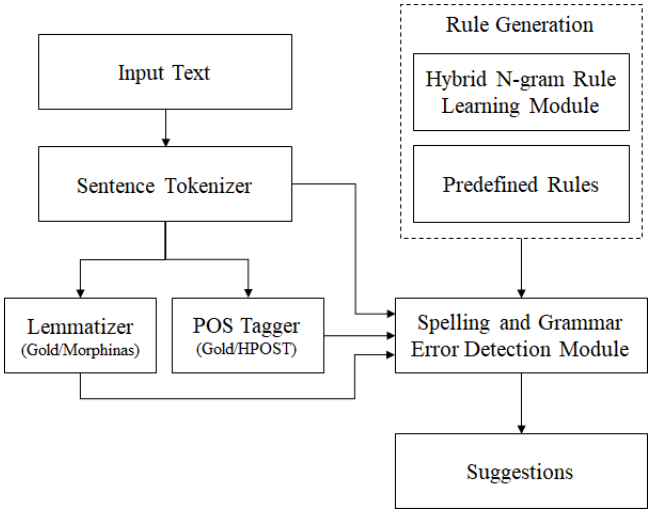

Fig. 1. System Framework of Gramatika

A. Rule Generation

Two resources are required by the spelling and grammar error detection module namely, the Hybrid N-Gram and Predefined Rules, which is used by Gramatika in determining which of the given sentences have errors and how to correct them. Moreover, these resources are generated beforehand and is only consulted by the spelling and grammar error detection module of Gramatika on runtime.

1) Hybrid N-Gram Rule Module: In inferring grammar rules, Gramatika follows an n-gram clustering approach having n-grams from lengths 4 to be converted into hybrid n-grams. The n-grams are considered one cluster if they have the same POS tags. If a cluster meets the minimum threshold of 4 n-gram members, it is then converted into a hybrid n-gram. If the word token in the same index in 75% of all the n-gram members are the same, token is retained as a word in the hybrid n-gram, else it is generalized into its POS tag. For instance, given the sample n-grams members *nagpunta sa bahay* 'went in the house', *namili sa bayan* 'bought in the city', the hybrid n-gram generated is *[VBTS] sa [NNC* '[PAST_VERB] in [COMMON_NOUN]'. This representation allows the grammar checker to understand that the tokens generalized to its POS level can be any past tense verb or common nouns, while the word *sa* 'in / at' is not freely substitutable by other conjunctions within the same POS tag group such as *o* 'or', and *pero* 'but'.

2) Predefined Rules: Manually-created rules were integrated inside Gramatika to see how it can work together with the hybrid n-gram rules. These manually-created rules uses regular expressions and a dictionary generated from a corpus focusing on minimal or no precision errors when applied. The list of rules are as follows:

- Use of *na* / *-ng* - If token *A* ends in a vowel or letter *n* and is followed by token B *na*, suggest to delete token B *na* and append *-ng* (if it ends with a vowel) or *-g* (if it ends with *n* to Token A).
  - ex. *pagkain na* ⇒ *pagkaing*
- Unmerging the token *mas* from verbs - If a token has the prefix *mas/Mas* and the remaining characters is an existing word in the dictionary, suggest an unmerging of the prefix *mas/Mas* as a separate token/word.
  - ex. *masmalakas* ⇒ *mas malakas*
- Merging Incorrectly Unmerged Affixes - Using the list of affixes from MagTag [2] a rule is created such that if an affix is seen as a token *A*, merge it with its succeeding token *B*. Only affixes that require no morphophonemic change when merged with a word was used by this rule.
  - ex. *pinag sikapan* ⇒ *pinagsikapan*
- Removing Incorrect Hyphenations - If a token has a hyphen, temporarily remove the hyphen and see if it is a word in the corpus. If yes, suggest removal of the hyphen.
  - ex. *pinaka-matalino* ⇒ *pinakamatalino*

B. Sentence Tokenizer

Before passing the given input text to Gramatika's major operations: POS tagging, lemmatizing, and spelling and grammar error detection, the given input is tokenized – a process functioning by separating words and punctuations into "tokens". An instance for this is provided the sentence "I am fine, thank you.", the words are separated into I, am, fine, ',', thank, you, '.' or to better visualize, transformed into "I am fine , thank you ." After running the text in this module, the input is ready for the major operations, starting from POS Tagging.

C. Hybrid Part-of-Speech Tagger (HPOST)

Aside from having one experiment in using a Gold standard for providing the POS tags, another utilizes HPOST, an enhanced successor of the Statistical Machine Translation Part-of-Speech Tagger (SMTPOST) [9] running on the same language domain, Filipino. It follows the MGNN Tagset[2] based from the same work, introducing the 'UNK' tag for unknown or untagged words. It includes a set of

[1]https://isipsafe-gramatika.appspot.com/

[2]http://goo.gl/dY0qFe

pre-processing modules for cleaning, tokenization and error prevention before execution of the SMT module. From the original feature extractor and statistical machine translation process, SMTPOST's tagging process has been extended using three rule-based approaches:

Dictionary Substitution Approach (DSA) utilizes the Word-Tag Dictionary from SMTPOST, searching for words within the dictionary and replacing them with their Part-of-Speech (POS) counterparts if there are matches. Entries were increased from 309 to 517 Word-POS pairs.

Regex-based Approach (RBA) functions the same way as DSA; instead of word searches, it uses regular expressions to replace feature patterns with tags. For example, given the feature *<*number*>, a corresponding regular expression (e.g. *[0-9]+) for cardinal numbers will match the feature, in turn replacing it with the CDB tag. Rules formulated are based from common patterns, mostly matching with numerical values (e.g. 100,000, 1st, 1990s, 10:30 and 02/14/17), abbreviations (e.g. PNP, Bb., et_al., and R&B), and [proper/common] nouns (i.e. token with no features or are whole words '*').

'UNK' (UNK) Tagger labels untagged words left by the statistical and rule-based taggers using the 'UNK' (Unknown) tag for consistency in the output.

Adding to the changes, SMTPOST's parallel corpus was increased from 2,668 to 15,166 sentences, adjusting both training (12,133) and testing (3,033) data while following the same 80:20 ratio. With SMT component alone, it's accuracy improved to 88.72%, close to a 4% difference from SMTPOST. Evaluating HPOST as a whole, it garnered the accuracy score of 91.38%, a 6.63% increase from SMTPOST and at least 10.38% higher than other POS taggers in the Philippines. A simulation of HPOST is provided below, starting from the given input sentence and ending with UNK module, delivering the final POS sentence counterpart:

- Input: Austria ang may pinakamataas na energy intake sa pagkain na humigit kumulang sa 3,800 kcal .
- FEX: :FS*austria #ang #may pa ka ma@in #na *energy i #sa pa #na #higit@um @um #sa *3,800 *kcal #.
- SMT: :FS*austria DTC VBH JJCS_JJD CCP FW FW CCT NNC CCP VBAF VBAF CCT *3,800 *kcal PMP
- DSA: :FS*austria DTC VBH JJCS_JJD CCP FW FW CCT NNC CCP VBAF VBAF CCT *3,800 NNCA PMP
- RBA: :FS*austria DTC VBH JJCS_JJD CCP FW FW CCT NNC CCP VBAF VBAF CCT CDB NNCA PMP
- UNK: UNK DTC VBH JJCS_JJD CCP FW FW CCT NNC CCP VBAF VBAF CCT CDB NNCA PMP

D. Morphinas Lemmatizer

Lemmatization is the process of removing inflections in a word. An example for this is the word *pinakamaganda* 'most beautiful', where the prefix *pinaka-* is removed, leaving the root *maganda* 'beautiful' as its output. To clarify, based on the framework the lemmatizer comes after the POS Tagger, but this only constitutes the order of process and does not take the tags as this module's input. Furthermore, a Gold standard of lemmatized words and a simple lemmatizer called *Morphinas* are used for this research.

Morphinas is a modified version of [11]'s Tagalog Wordframe model. It utilizes an extensive list of possible prefixes and substring combinations, particularly an increased size in [11]'s database of root words. It was tested through 460 sentences with 13,587 words (3,466 unique) and garnered 65% accuracy.

E. Spelling and Grammar Error Detection

In detecting spelling and grammatical errors and producing suggestions based on the hybrid n-grams, a weighted Levenshtein edit distance algorithm is used. Gramatika extends the error detection and correction of EdIt [4] and Lexbar [13] differentiating substitution types, proposing new suggestion types: unmerging and merging (see Table I). Prioritization based on error types are also defined as follows, 1 being the top priority: 1. Incorrect Affix/Form, 2. Spelling Error, 3. Incorrectly Merged and Incorrectly Unmerged, 4. Wrong Word but Same POS Group, 5. Wrong Word, 6. Missing Words, and 7. Unnecessary Words.

Correction	Error Description
Substitution	Affix or Form errors, wrong word or punctuation usage
Spelling Correction	Misspelled words, misuse or lack of hyphens
Insertion	Missing words and punctuations
Deletion	Unnecessary words and punctuations
Unmerging	Incorrectly merged words requiring unmerging of words or removal of hyphens
Merging	Incorrectly unmerged word requiring removal of space or insertion of hyphen between texts

TABLE I
ERROR CORRECTION TYPES

F. Suggestions

After Gramatika's major operations, its output returns a list of suggestions including the following information: input text, POS tags and lemma counterparts, edit distance value, and list of spelling and/or grammar suggestions.

III. RESULTS & ANALYSIS

To generate hybrid n-gram rules, Gramatika is trained on a corpus of 7,384 manually POS-tagged and lemma-tagged, complex, error-free Filipino sentences which consists of 183,533 tokens (25,674 unique), and tested on both erroneous and error-free data. A total of 248 erroneous phrases were retrieved from English-to-Filipino translation exercises containing spelling and grammatical errors written by second year university students and a total of 1,284 sentences were annotated, validated, and manually-tagged with its respective POS tags and lemmas by two Filipino linguists from the same university. Three types of tests

were performed: (1) using gold standard (manually tagged) POS and gold standard lemmas on test inputs (G-G), (2) using HPOST POS tagger-generated POS tags and gold standard lemmas (PT-G), and (3) using gold standard POS tags and Morphinas lemmatizer-generated lemmas (G-L). The evaluation metric used for error detection and correction suggestion is based from [12] that separately counts detected errors with correct suggestions, detected errors with incorrect suggestions, and undetected errors / absent suggestions.

Tests G-G and G-L produced the same results in both erroneous phrases and error-free because despite Morphinas scoring a lemmatization accuracy of 67% and 76.56% on the respective test datasets, expected suggestions were still produced since the lemmas are only used for prioritizing 'Wrong Word Form' correction/s among others. Three instances were observed wherein because of incorrect lemmas, the expected suggestions were ranked as misspelling-type suggestions but still found within the top 2 suggestions. On error-free texts, lemmas are considered irrelevant due to the nature of the hybrid n-gram rules which only uses words and POS tags.

As seen in Table II, using gold standard POS tags led to a high error detection rate of 89.92%. This is primarily because the linguist tags misspelled words, incorrectly affixated words, incorrectly merged and unmerged words with an *unknown* POS tag [?]. The checker using G-G was able to produce correct suggestions 64.11% of the time indicating that the ideal suggestion are found within the top 2 suggestions produced by the checker.

Score Type	G-G	PT-G
Correct Suggestions	159 / 248 (64.11%)	101 / 248 (40.72%)
Incorrect Suggestions	64 / 248 (25.81%)	30 / 248 (12.10%)
Absent Suggestions	25 / 248 (10.08%)	117 / 248 (47.18%)
On Error-Free Texts	911 / 1284 (70.95%)	1091 / 1284 (84.97%)

TABLE II
RESULTS - ERRONEOUS PHRASES

Error Type	G-G	PT-G
Wrong Word Form	41 / 62	20 / 62
Mispelling	38 / 62	6 / 62
Incorrectly Unmerged Words	33 / 40	32 / 40
Incorrectly Merged Words	19 / 32	15 / 32
Wrong Word Usage	4 / 20	4 / 20
na / -ng	21 / 21	21 / 21
Missing Word or Punctuation	0 / 6	0 / 6
Extra Word or Punctuation	3 / 5	3 / 5

TABLE III
BREAKDOWN OF ERROR TYPES

Table III shows the list of error types that were found in the test data. Many of the errors are Wrong Word Form errors which are words that contain incorrect affix/es (ex. *ginagabay* , *ipapayagan*, and *binubukas* which should be *ginagabayan* 'being guided', *papayagan* 'will allow', and *binibuksan* 'opening', respectively) and incorrect placement / absence of hyphens (ex. *pinagaaralan, nagapply, tagtuyo* which should be *pinag-aaralan, nag-apply, tagtuyo* separating the prefix with a hyphen if the root starts with a vowel or if the root is an English word). Gramatika G-G scored 41 / 62 (66.13%) in Wrong Word Form errors followed by Misspelling errors in which G-G scored 38 / 62 (61.3%).

Analysis show that majority of the shortcomings of the grammar checker is caused by: the absence of hybrid n-gram rules needed to correct an erroneous phrase or flag a phrase as error-free, and absence of words in the dictionary (ex. *naglalakad* 'walking') to produce word-level suggestions for errors (ex. misspelled *nglalakad*). Both causes can be traced back to lack of training data, showing that such approach is corpus size dependent. Additionally, there were some errors that cannot be handled by the grammar checker since it requires 2 edit types. For instance, the word *masmaliwanang* cannot be corrected as *mas maliwanag* 'brighter' because it requires unmerging the word *mas* '-er' and a spelling correction from *maliwanang* to *maliwanag* 'bright'.

The tests conducted also highlighted the need to include plurality features in verbs, adjectives, and adverbs. Currently, the MGNN tagset does not consider the plurality of the words in these POS tag groups. For instance, the grammar checker still suggested a correction for the phrase *sa mga matas na bahay* 'in the *matas* houses' to change the misspelled word *matas* as *mataas* 'tall (singular)' as it requires a [JJD] (adjective) in *matas*'s word slot as the ideal choice *matataas* 'tall (plural)' is not in the dictionary. A certain test instance also highlights the need to distinguish pronouns into their three dimensions: *Panauhan* 'point-of-view', *Kailanan* 'number' and *Kaukulan* 'case'.

Some test instances also showed the limitation of the proposed approach to only perform 1 action in each suggestion. For the test input *o pang probinsya*, it was not able to merge the affix *pang* with the root *probinsya* as *pamprobinsiya* 'for provincial use' as it requires 2 actions: a merge and a spelling correction to apply the morphophonemic assimilation. Another test input *na may masmaliwanang* also required 2 actions to correct *masmaliwanang* as *mas maliwanag* 'brighter': an unmerge and spelling correction.

The manually-created rules was proven to be useful contributing 36 correct suggestions aiding the error detection for Mispellings with 3, Incorrectly Unmerged Words with 9, Incorrectly Merged Words with 3, and *na / -ng* with 21.

As seen in Table II and Table III, the results of G-G and PT-G has a large difference in terms of accuracy especially on Wrong Word Form and Misspelling error types. This is mainly because the linguist has manually tagged all misspelled words and incorrectly affixated words with the unknown POS tag [?] while the POS tagger, being only

trained using error-free data, produced a tagging accuracy of 70% only producing 25 out of the 168 expected unknown [?] POS tags. Such limitation suggests the need of developing tools to detect incorrect affix errors in Filipino and spelling errors that can be used by POS taggers as a preprocessor when needed. Training the POS tagger on a mix of error-free and erroneous data is also suggested for it to produce more unknown [?] POS tags. PT-G also performed higher (84.97%) on error-free texts than G-G (70.95%). Analysis show that consistency in tagging by a POS tagger compared to manual tagging of a human expert is important which led to the observed results.

IV. Summary & Future Works

This paper shows initial experiments in developing a grammar checker system (Gramatika) that covers a wide variation of error types found in the Filipino language using the available resources and linguistic tools (POS tagger and lemmatizer). Gramatika performed 64% accuracy in producing the expected suggestions using gold standard erroneous training data, and 40.72% when using POS tags generated by the state-of-the-art POS tagger in Filipino – HPOST. Whereas on error-free words, Gramatika scored 85% using HPOST's POS tags as data. Overall, the results show that the approach provides a huge potential in error detection as well as error correction, and will improve its results when given more resources – tagged error-free data and erroneous data. Aside from that, it is suggested to implement linguistic tools such as constituency parser, incorrect affix detection system, and a spell checker for the high-morphology Filipino language to further greaten the capabilities of Gramatika.

Acknowledgments

This research work is funded by Philippine Department of Science and Technology through its Interdisciplinary Signal Processing for Pinoys: Software Applications for Education (DOST:ISIP-SAFE) research program and is supported by De la Salle University – Manila.

References

[1] M.J. Alam, N. UzZaman, and M. Khan, "N-gram based statistical grammar checker for Bangla and English," IEEE Trans. Visualization and Computer Graphics, 2006.

[2] M. Aquino, E. Fernandez and K. Villanueva (2007), "Mag-tagalog: A Rule Based Tagalog Morphological Analyzer and Generator," B.S. Thesis, De La Salle Univ., MM, 2007.

[3] M.P. Go and A. Borra, "Developing an Unsupervised Grammar Checker for Filipino Using Hybrid N-grams as Grammar Rules," Proc. 30th Pacific Asia Conference on Language, Information and Computation, 2016, pp. 105113.

[4] C. Huang, M.H. Chen, S.T. Huang, and J. Chang, "EdIt: A broad-coverage grammar checker using pattern grammar," Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 2631.

[5] N.Y. Lin, K.M. Soe, and N.L. Thein, "Developing a chunk-based grammar checker for translated English sentences," Proc. 25th Pacific Asia Conference on Language, Information and Computation, 2011, pp. 245254.

[6] C. Manning, "Part-of-speech tagging from 97% to 100%: Is it time for some linguistics?," Proc. 12th International Conference on Intelligent Text Processing and Computational Linguistics, 2011, pp. 171189.

[7] R. Nazar and I. Renau, "Google books n-gram corpus used as a grammar checker," Proc. EACL 2012 Workshop on Computational Linguistics and Writing, 2012, pp. 2734.

[8] D. Nicholls, "The Cambridge Learner Corpus error coding and analysis for lexicography and ELT," Cambridge University Press, 1999.

[9] N. Nocon and A. Borra, ""SMTPOST: Using Statistical Machine Translation Approach in Filipino Part-of Speech tagging," Proc. 30th Pacific Asia Conference on Language, Information and Computation, 2016, pp. 391-396.

[10] A. Rozovskaya and D. Roth, "Building a state-of-the-art grammatical error correction system," In Trans. Association of Computational Linguistics, vol. 2, pp. 414434, 2014.

[11] C. Cheng and S. See, "The revised wordframe model for the Filipino language," Journal of Research for Science, Computing and Engineering, vol. 3, pp. 17-23, 2006.

[12] M. Starlander and A. Popescu-Belis, "Corpus-based Evaluation of a French Spelling and Grammar Checker," Proc. 3rd International Conference on Language Resources and Evaluation, 2002, pp. 268274.

[13] N.L. Tsao and D. Wible, "A method for unsupervised broad-coverage lexical error detection and correction," Proc. NAACL HLT Workshop on Innovative Use of NLP for Building Educational Applications, 2009, pp. 5154.