Journal of Advances in Mathematics and Computer Science

23(3): 1-15, 2017; Article no.JAMCS.32927

Previously known as British Journal of Mathematics & Computer Science
ISSN: 2231-0851

Modelling Grammaticality-grading in Natural Language Systems Using a Vector Space Approach

Moses Kehinde Aregbesola[1*], Rafiu Adesina Ganiyu[2], Stephen Olatunde Olabiyisi[2], Elijah Olusayo Omidiora[2] and Oluwaseun Olubisi Alo[2]

[1]Department of Mathematics and Computer Science, Elizade University, Ilara-Mokin, Nigeria.
[2]Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

Authors' contributions

This work was carried out in collaboration between all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2017/32927
Editor(s):
(1) Sun-Yuan Hsieh, Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan.
Reviewers:
(1) S. K. Srivatsa, Prathyusha Engg College, India.
(2) Tony (Anthony) Spiteri Staines, University of Malta, Malta.
Complete Peer review History: http://www.sciencedomain.org/review-history/20090

Received: 23rd March 2017
Accepted: 12th April 2017
Published: 17th July 2017

Original Research Article

Abstract

There exist several natural language processing systems that focus on checking the grammaticality (grammatical correctness or incorrectness) of natural language texts. Studies however showed that most existing systems do not assign specific scores to the grammaticality of the analysed text. Such scores would for instance prove very useful to second language learners and tutors, for judging the progress made in the learning process and assigning performance scores respectively. The current study was embarked upon to address this problem. A grammaticality grading model which comprised of 6 equations was developed using a vector space approach. The model was implemented in a natural language processing system. Correlation analysis showed that the grading (in %) performed using the developed model correlated at a coefficient of determination (R^2) value of 0.9985 with the percentage of grammatical sentences in evaluated texts. The developed model is therefore deemed suitable for grammaticality grading in natural language texts. The developed model would readily find use in computer aided language learning and automated essay scoring.

**Corresponding author: E-mail: kennie747@yahoo.com, kehinde.aregbesola@elizadeuniversity.edu.ng;*

Keywords: Computational linguistics; automated evaluation; grammar; grammaticality; grammaticality grading; natural language processing; Vector Space Model; Mosesean Vector Space.

1 Introduction

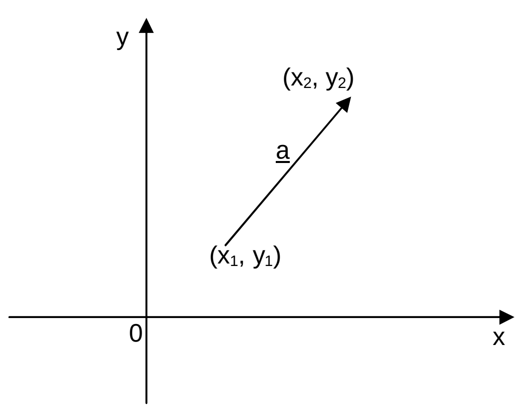
A number of grammar-checkers systems exist that focus on checking the grammaticality (grammatical correctness or incorrectness) of natural language texts [1]. Studies however showed that most existing systems do not assign specific scores to the grammaticality of the analysed text [2]. Such scores would for instance prove very useful to second language learners and tutors, for judging the progress made in the learning process and assigning performance scores respectively. The current study was embarked upon to address this problem. English (Formal Standard English) was the language adopted in this study.

2 Vectors and Vector Spaces

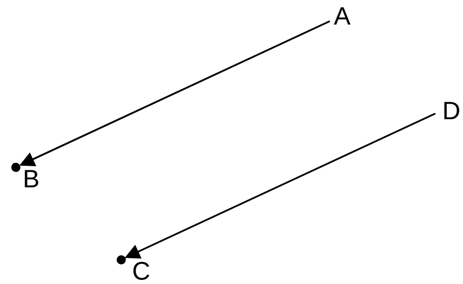
Vectors and vector spaces are closely related terms as the latter is dependent on the former. Vector quantities are generally defined as quantities that have both magnitude and direction, unlike scalar quantities which have only magnitude but no direction [3]. A vector space on the other hand is a set of vectors along with operations of addition and multiplication such that the set is a commutative group under addition, having a multiplicative inverse, and including multiplication by scalars which is both associative and distributive [4,5,6].

2.1 Vectors

Vectors and points are common data structure considered in many areas of Mathematics and Computer Science. They are applied extensively in data compression, image processing, computer vision, computer graphics, and numerical analysis. Two-dimensional vectors can be defined as directed arrows in the plane. The position of the arrow is not important. The length (magnitude) and direction of the arrow are the important features of the vector, and they determine the vector. They can be added, scaled and rotated [7]. Vectors having the same length and direction are said to be equivalent [3]. Two vector in the same direction are said to be parallel. The zero vector has a magnitude of zero and is denoted as $\underline{0}$. Fig. 1(a) shows a vector $\underline{a}$ between two points (x_1, y_1) and (x_2, y_2). In Fig. 1(b), the vectors $\underline{AB}$ and $\underline{DC}$ are equivalent, because two-dimensional vectors are distinguished only by length and direction. They are thus treated as equal i.e. $\underline{AB} = \underline{DC}$.



(a)



(b)

Fig. 1. (a) Vector in the plane (b) Two equivalent and parallel vectors
(Lindeman, 2008; Kambites, 2014)

2.2 Vector space

The concept of vector space has been the central point of discussion of a number of literatures [4,5,6,8,9,10, 11,12]. A vector space V can summarily be defined as a set of vectors over the field F (such as real, complex, and natural numbers) which may be added together and multiplied (or scaled) by numbers referred to as scalars, such that for all $\underline{x}$, $\underline{y}$, $\underline{z} \in V$, the following eight axioms are satisfied:

i. Associativity of addition.
   $\underline{\mathbf{x}} + (\underline{\mathbf{y}} + \underline{\mathbf{z}}) = (\underline{\mathbf{x}} + \underline{\mathbf{y}}) + \underline{\mathbf{z}}$;
ii. Commutativity of addition. $\underline{\mathbf{x}} + \underline{\mathbf{y}} = \underline{\mathbf{y}} + \underline{\mathbf{x}}$;
iii. Identity element of addition. There exists an element $\underline{\mathbf{0}} \in V$, called the zero vector, such that $\underline{\mathbf{y}} + \underline{\mathbf{0}} = \underline{\mathbf{y}}$ for all $\underline{\mathbf{y}} \in V$;
iv. Inverse element of addition. For every $\underline{\mathbf{y}} \in V$, there exists an element $-\underline{\mathbf{y}} \in V$, called the additive inverse of $\underline{\mathbf{y}}$, such that
   $\underline{\mathbf{y}} + (-\underline{\mathbf{y}}) = \underline{\mathbf{0}}$;
v. Compatibility of scalar multiplication with field multiplication. $a(b\underline{\mathbf{y}}) = (ab)\underline{\mathbf{y}}$;
vi. Identity element of scalar multiplication.
   $1\underline{\mathbf{y}} = \underline{\mathbf{y}}$, where 1 denotes the multiplicative identity in F;
vii. Distributivity of scalar multiplication with respect to vector addition.
   $a(\underline{\mathbf{x}} + \underline{\mathbf{y}}) = a\underline{\mathbf{x}} + a\underline{\mathbf{y}}$; and
viii. Distributivity of scalar multiplication with respect to field addition. $(a + b)\underline{\mathbf{y}} = a\underline{\mathbf{y}} + b\underline{\mathbf{y}}$

[5,6,8,9,10].

Depending on the literature, these expressions are sometimes compressed to give fewer axioms or expanded to give more axioms, expressing the exact same concepts.

2.3 NLP applications of vectors and vector spaces

The concept of vector space is considered in most linguistic and NLP literatures from the perspective of lexical and semantic distribution. Semantic vector space models of language make use of real-valued vectors to denote each word that are typically associated with a particular word. Words that typically occur together are assigned values that often depict their probability of occuring together in a sentence. They are used for a wide range of NLP operations including grammaticality evaluation and error detection. Detailed semantic and syntatic regularity have been successfully captured using vector space representations and vector arithmetic. The study of Pennington, Socher and Manning [13] came up with global vectors for word representation (GloVe). The study focused on highlighting the properties that made the emergence of such captured regularities possible in word vectors. Schmid [14] however focused on efficient parsing of highly ambiguous context-free grammars using bit vectors.

The study of Stolcke [15] represented a formalism dubbed Vector Space Grammars (VSG) for deriving phrase structure categories that made use of structured samples of a context-free language. Using the connectionist approach, the entire training process made use of adaptation, competition and error back-propagation, all occurring in a continuous vector space. It advocates the use of vectors instead of symbols for the purpose of linguistic category labeling.

Vector Space Semantic Parsing (VSSP) presented in the work of Krishnamurthy and Mitchell [16] is a framework for learning compositional models of vector space semantics. It applies combinatory categoria grammar (CCG) to define the relationship between syntactic categories and semantic represenatations, taken as vectors and functions on vectors. Using CCG based semantic parser, texts are parsed into lambda calculus formulae that compute to equivalent vector space representations.

In general, vector space models make use of vectors and operations on vectors to represent the semantics of natural language expressions [17]. A number of other studies including Coecke, Grefenstette and Sadrzadeh [18], Socher, Pennington, Huang, Andrew and Manning [19], Socher, Huval, Manning and Ng [20], Turney [21] and Rapp [22] focused on similar concepts. The studies achieved significant performances that corresponded well with human judgment.

Grammaticality is considered a vector concept within this literature, having both magnitude and direction. This is in contrast with scalar quantities that have only magnitude but no direction. The direction of grammaticality is either towards grammatical correctness, or away from grammatical correctness. Grammatical correctness is a state described as Grammatical Equilibrium (GE) within this study, and is ascribed a gradience value of zero (0).

On a general note, grammars are usually designed to express the state of grammatical equilibrium. For constraint based grammars [23,24], each appropriate constraint within the grammar enforces the grammar towards being able express or determine sentences that are grammatically correct.

Furthermore, although grammaticality [25] is generally used to express the state of grammatical correctness or otherwise of a sentence, it is sometimes used strictly as a measure of grammatical correctness, especially when used alongside ‘ungrammaticality’. From this perspective, grammaticality is used as a measure of grammatical correctness while ungrammaticality is used as a measure of grammatical incorrectness.

3 Formulation of a Vector Space Model for Grammaticality Grading

The formulation of a vector space model for grammaticality grading is presented in this paper. The approach employed in the current study is similar to that of Aregbesola et al. [26] but with enhancements made to the developed model to account for situations with no ungrammatical sentence in the evaluated texts. The current study further went ahead to implement and validate the formulated model. The formulation is in three phases: (i) The vector space V of grammaticality vectors is defined; (ii) The resultants of these vectors are shown to lie within the vector space; and (iii) A set of grammaticality gradience equations are derived using the formulated vector space for grammaticality grading.

3.1 Definition of the vector space (V) of grammaticality

Let $\underline{x}$, $\underline{y}$, $\underline{z}$ be weighted entities associated with grammaticality such as possible error categories. Such error categories include missing-word, extra-word, real-word spelling, verb-form, punctuation and agreement errors, which are uniquely identifiable within a sentence. Also let $\{\underline{x}, \underline{y}, \underline{z}, ...\} \in V$. Like any other standard vector space, V is a set of vectors over the field F (which in this case is the set of real numbers R) which may be added together and multiplied (scaled) by numbers referred to as scalars, such that the eight axioms listed in Section 2.2 are satisfied.

As depicted in Figs. 2, 3 and 4, these grammaticality vectors forthwith dubbed Mosesean vectors are linear (one-dimensional) in nature. Therefore, addition operations on these vectors are by simple arithmetic summation. Furthermore, ungrammatical elements within input sentences are assigned negative values; while counter measures to correct such ungrammaticality are assigned positive values. Thus, the magnitude of grammaticality the Mosesean vectors introduce into the system at any point in time is totally dependent on the magnitude of existing ungrammaticality. Figs. 2, 3 and 4 illustrate these concepts.

3.2 Resolving the resultants of Mosesean vectors

The default value of zero (0) is assigned as gradience value to any sentence introduced into the system of Mosesean vectors. At this default value, the sentence is at equilibrium, and is completely grammatical. This equilibrium is toppled when ungrammatical elements are identified within the sentence. When a sentence is ungrammatical by a certain magnitude, the system attempts to find complementary grammaticality measures to pull the sentence back into equilibrium as depicted in Fig. 4. Grammaticality (+g) is generated in response to Ungrammaticality (-g). +g can only be as large as to cancel out -g, resulting in equilibrium.

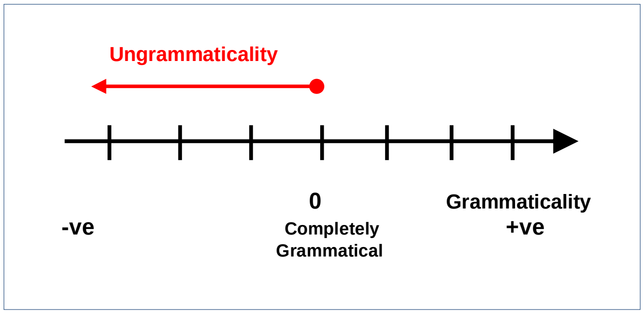


Fig. 2. Mosesean vectors on a real number line showing –ve Grammaticality ≡ Ungrammaticality

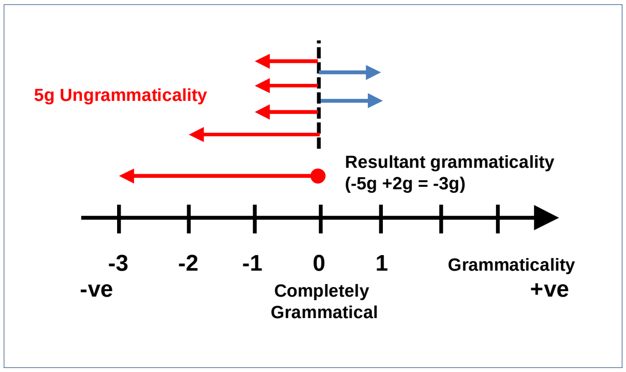


Fig. 3. Resultants of Mosesean vectors not yet at equilibrium

3.3 Computing grammaticality gradience using Mosesean vectors

The default value of zero (0) is assigned as gradience value to any sentence introduced into the system of Mosesean vectors. If the sentence is grammatically correct, the gradience remains unchanged at zero and requires no further computation. However, if ungrammaticality (-g) is found within the sentence, the cumulative ungrammaticality ($-G_{sum}$) is the arithmetic sum of the individual ungrammaticality values.

Thus:

Let g_0 be the default case (for a grammatical sentence) with no error, therefore $g_0 = 0$.

If the values assigned to ungrammaticality items in a sentence are $-g_1, -g_2, -g_3, \ldots -g_n$,

where n is the number of ungrammaticality items in the sentence,

then

$$-G_{sum} = g_0 + (-g_1) + (-g_2) + (-g_3) + \ldots + (-g_n)$$

$$-G_{sum} = \sum_{i=1}^{n} -g_i + g_0 \tag{1}$$

If only the magnitudes are considered, ignoring the signs, then (3.1) becomes

$$G_{sum} = \sum_{i=0}^{n} |g_i|$$

Therefore, the cumulative grammaticality (G_{sum}) required to bring the sentence into equilibrium is:

$$G_{sum} = \sum_{i=0}^{n} |g_i| \tag{2}$$

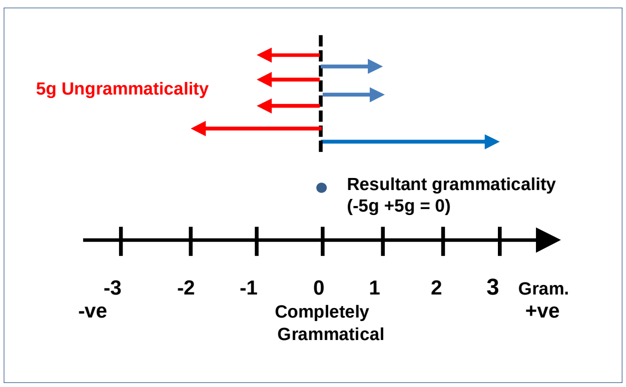


Fig. 4. Resultant of Mosesean vectors at equilibrium

Furthermore in the system, the grammaticality gradience (μG) of an ungrammatical sentence is computed by dividing the cumulative grammaticality (G_{sum}) by the total number of leaf nodes in the sentence parse tree. Since the number of leaf nodes in a sentence parse tree is always equal to the number of words in the sentence, it therefore follows that grammaticality gradience (μG) for a sentence of length m is:

$$\mu G = \frac{\sum_{i=0}^{n} |g_i|}{m} \tag{3}$$

Assigning a value of magnitude one (1) to each ungrammaticality item in an input sentence, it follows that the cumulative grammaticality (G_{sum}) can at most be as large as the number (m) of words in the sentence.

Thus from equation (3.3):

when $n \rightarrow 0$:
$m = m$

and $\mu G \rightarrow 0$

when $n \rightarrow \infty$:
$m \rightarrow \infty$

and $\mu G \rightarrow 1$

Hence, the grammaticality gradience μG is such that:

$$0 \leq \mu G \leq 1 \tag{4}$$

As grammaticality evaluation extends beyond the evaluation of a single sentence to the evaluation of multiple sentences, the gradience for each sentence is computed in the same manner, applying (3.3) to each of them. The arithmetic mean of the gradience(s) of the different sentences is then computed to give the gradience of the entire text (all the sentences put together).

Thus:

If there are q sentences with gradience values μG_1, μG_2, μG_3, … μG_q,

then

$$\mu G_{sum} = \mu G_1 + \mu G_2 + \mu G_3 + \ldots + \mu G_q$$

$$\mu G_{sum} = \sum_{i=1}^{q} \mu G_i \tag{5}$$

and by extension, the mean grammaticality gradience μG for multiple sentences is:

$$\mu G = \frac{\sum_{i=1}^{q} \mu G_i}{q} \tag{6}$$

Finally, since grammaticality gradience (μG) is a measure of deviation from grammaticality, the actual Grammaticality Score (GS) of a text would be evaluated by subtracting the gradience value from 1 and multiplying by 100 percent.

Thus:

$$GS = (1 - \mu G) \times 100\% \tag{7}$$

3.4 Illustrations

This section illustrates how the formulated vector space model for grammaticality grading woks. Three different cases are considered. The first case illustrates computation for a grammatical sentence. The second case illustrates computation for an ungrammatical sentence. The third case illustrates computation for multiple sentences comprising of grammatical and ungrammatical sentences.

3.4.1 Case 1: Grammatical sentence

Consider the grammatical sentence "The fat pony sleeps in the barn". To evaluate this sentence for grammaticality, HPSG is employed in a bottom-up fashion as shown in the parse tree in Fig. 5. As each lexical token in the sentence is parsed upward in the parse tree, it is replaced by its corresponding feature structure. For reasons of convenience, the parse tree in Fig. 5 only shows the POS components of the respective feature structures for each lexical entry. Since the sentence was successfully parsed all the way to the topmost root node (S in Fig. 5), and no error (n=0) was identified during the parse process. The sentence is therefore considered grammatical. The grammaticality gradience for this particular sentence is therefore computed as follows:

Using equation (3.3)

where n = 0 and m = 7

$$\mu G = \frac{0}{7}$$
$$\mu G = 0$$

Therefore, the Grammaticality Score (GS) from equation 3.7 is given as

$$GS = (1 - 0) \times 100\%$$
$$GS = 100\%$$

which is the value expected for a grammatical sentence.

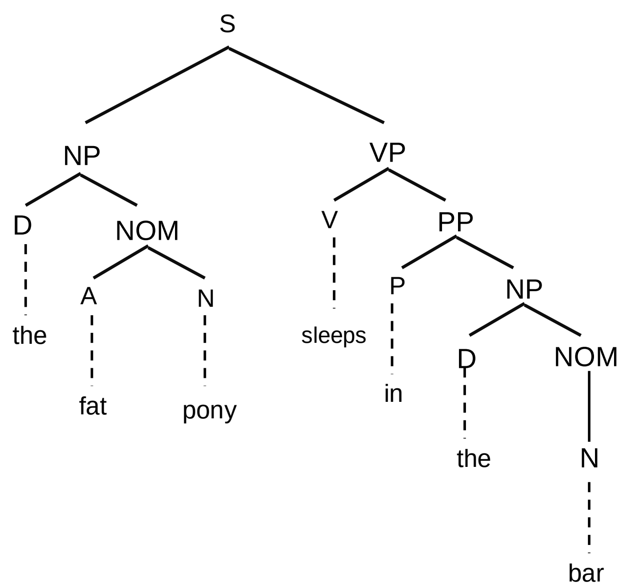


Fig. 5. A phrase structure tree for "The fat pony sleeps in the barn" [27]

3.4.2 Case 2: Ungrammatical sentence

Consider the ungrammatical sentence "I loves Sandy". The simplified feature structure for each lexical entry is shown in Fig. 6. Fig. 7 shows the agreements expected of the lexical entries. The numbers in square bracket (e.g. [1]) show what attributes should agree. Comparing the agreement between "I" and "loves", a subject-verb error was observed (n = 1). "I" is first (1) person singular (sg) and therefore expects a first person singular verb, but rather it gets a third (3) person singular verb "likes" and vice versa. The ARG-ST feature for the verb "loves", < loves, [ARG-ST < [NP [AGR 3s]], NP >] > shows that the third lexical entry "Sandy" is in agreement with the object expected by the verb.

Thus, to compute the grammaticality gradience for this particular sentence:

Using equation (3.2)

where $n = 1$ and $g = 1$
$G_{sum} = 1$

Using equation (3.3)

where $n = 1$ and $m = 3$

$$\mu G = \frac{1}{3}$$
$$\mu G = 0.3333$$

Therefore, the Grammaticality Score (GS) from equation 3.7 is given as

$$GS = (1 - 0.3333) \times 100\%$$
$$GS = 66.67\%$$

which is within the range of values expected. It should be noted that the feature structures shown in Fig. 6 and Fig. 7: are highly simplified.

(a) $$\begin{bmatrix} \text{POS} & \textit{pronoun} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{PERS} & \textit{1} \end{bmatrix} \end{bmatrix}$$

(b) $$\begin{bmatrix} \text{POS} & \textit{verb} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{PERS} & \textit{3} \end{bmatrix} \end{bmatrix}$$

(c) $$\begin{bmatrix} \text{POS} & \textit{noun} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{PERS} & \textit{3} \end{bmatrix} \end{bmatrix}$$

Fig. 6. (a) AVM for the word "I" (b) AVM for the word "loves" (c) AVM for the word "Sandy"

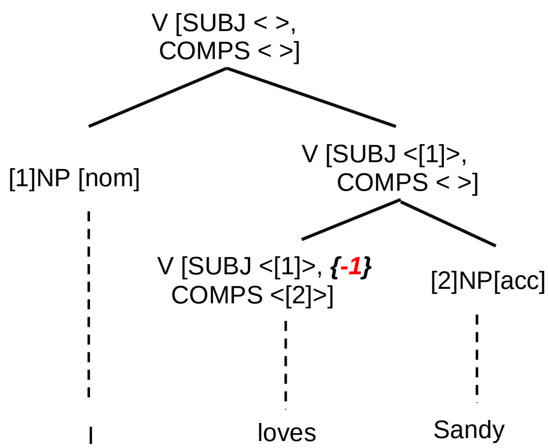


Fig. 7. Parse tree for the sentence "I loves Sandy" highlighting the ungrammaticality weight

3.4.3 Case 3: Multiple sentences

Finally, consider a text that consists of both sentences in illustration 1 and 2. That is, "The fat pony sleeps in the barn. I loves Sandy". To compute the overall grammaticality gradience for the text, the gradience for the individual sentences is computed as has already been done in illustrations 1 and 2. Equation (3.6) is then used to compute the overall grammaticality of the text as follows:

Using equation (3.6)

where $q = 2$ and $\mu Gi = \{0, 0.3333\}$

$$\mu G = \frac{0 + 0.3333}{2}$$

$$\mu G = 0.1667$$

Therefore, the Grammaticality Score (GS) from equation 3.7 is given as

$$GS = (1 - 0.1667) \times 100\%$$

$$GS = 83.33\%$$

which is also within the range of values expected for a text consisting of both grammatical and ungrammatical sentences.

4 Implementing the Model

The model was implemented in a natural language processing system described in the work of Aregbesola, Ganiyu, Olabiyisi and Omidiora [28]. The Grammaticality Evaluation Systems (GES) which is a computer-based natural language processing system is used to classify natural language texts as either grammatical or ungrammatical. The GES was developed using the constraints-based approach of Handcrafted Grammar (HG).

Sample texts of grammatical sentences used in the study were acquired from different sources including the British National Corpus, language experts and print media. Ungrammatical sentences were generated from the grammatical texts via word substitution, insertion and deletion. The grammaticality evaluation system architecture was designed based on the HG formalism. The system architecture was implemented using Visual Studio .NET and OpenNLP statistical lexical parser. The performance of the implemented system was evaluated with sets of grammatical and ungrammatical texts containing 1780 sentences each. The performance metrics employed in the study include Precision, Recall, Accuracy, F1-Score, False Positive Rate and Execution Time. The relationship between Average Execution Time (AET) and Number of Words (NW) per sentence was equally investigated.

The evaluation results in Table 1 showed that the system yielded Execution Time, Precision, Recall, Accuracy, F1-Score and False Positive Rate values of 0.0409 second, 0.9091, 0.9831, 0.9424, 0.9447 and 0.0983, respectively. The investigation of the relationship between AET and NW revealed that the AET per sentence was directly proportional to the NW in the sentence. The study resulted in a system that evaluated the grammaticality of English language texts and also assigned specific percentage score values to the grammar of the evaluated text.

Table 1. Overview of the systems' performance [28]

Measures	Value
Precision	0.9091
Recall	0.9831
Accuracy	0.9424
F1 score	0.9447
False positive rate	0.0983
Execution time (seconds)	0.0409

5 Validating the Model

The validity of the developed model was tested using 1780 grammatical sentences and 1780 ungrammatical sentences, making a total of 3560 sentences. These sentences were collected from the British National Corpus [29,30], language experts, as well as from other online and print media sources. The BNC was chosen because it was designed to represent a wide cross-section of British English, both spoken and written, from the late twentieth century. The corpus of ungrammatical texts was generated from the corpus of grammatical texts by word substitution, insertion and deletion, an approach similar to that of Foster [31,32].

Using the formulated Mosesean Vector Space Model in a natural language processing system, the average Grammaticality Scores (GS) for the considered grammatical and ungrammatical sentences were determined. Texts which consist of 0 - 50%, 50 - 75%, and 75 - 100% grammatical sentences were classified as Poor (P), Average (A) and Good (G), respectively. The ranges of GS characterising P, A and G sentences were then computed. The Coefficient of Determination (R^2) describing the fitness of the formulated vector space model was also computed.

The average GS for the considered grammatical and ungrammatical sentences were 99.64 and 73.34%, respectively. The GS characterising P, A and G sentences were in the ranges ($0 \leq GS < 87\%$), ($87 \leq GS \leq 94\%$) and ($94 < GS \leq 100\%$), respectively. This information is shown in Table 1.

The regression chart of Fig. 8 shows the analysis on the data points extracted from the data ranges in Table 1. The analysis yielded a Coefficient of Determination (R^2) value of 0.9985. The relationship between the systems assigned score (y) and the percentage (x) of grammatical sentences in the texts is shown in the regression equation (4.1).

$$y = 0.2656\text{x} + 73.551 \tag{8}$$

Table 2. Analysis of grammar scores

Data ranges		Data points	
% Range of grammatical sentences in text	**% Range of scores assigned by the model**	**Grammatical sentences in text (%)**	**Scores assigned by the model (%)**
$0 < x \leq 50$	$73.34 < y < 87$	0	73.34
$50 < x \leq 75$	$87 \leq y \leq 94$	50	87
$75 < x < 100$	$95 \leq y < 99.64$	75	94
		100	99.64

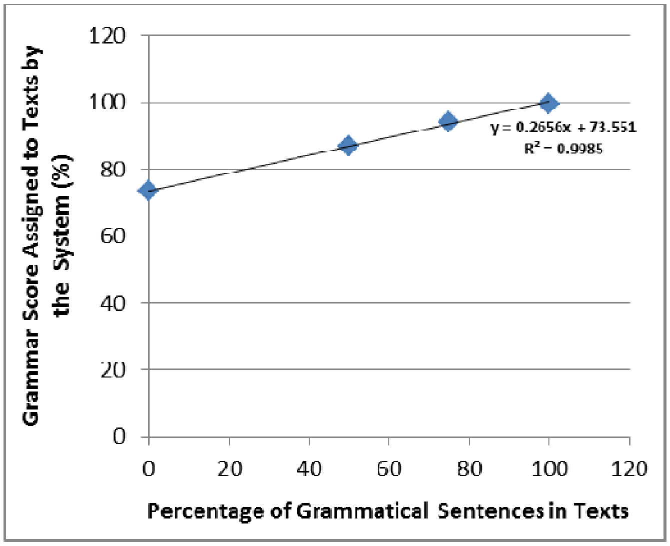


Fig. 8. Regression chart of system-assigned grammar scores (in %) against percentage grammatical sentences in texts

6 Conclusion

This research was embarked upon to meet the need for a grammaticality grading model that make it possible for systems to provide graded grammaticality evaluation feedback to users and learners of English language. In order to be able to provide the required graded feedback, there was the need for a grading mechanism. A vector space model for grammaticality grading was developed and subsequently implemented as the grammaticality grading mechanism for an existing grammar-checker system.

The developed model for grammaticality grading which was successfully integrated into an existing grammar-checker system effectively assigned specific percentage grammar scores to texts entered into the system. The average execution time (0.0409 seconds/sentence) of the system implementing the model was not noticeably altered. Hence, the developed grammar-grading model is very fast and does not constitute an overhead with respect execution time.

Regression analysis between the grammar-score assigned by the system and the percentage of grammatical sentences in input texts showed a Coefficient of Determination (R^2) value of 0.9985, meaning that the data is very close to the fitted regression line. Hence, the developed Mosesean Vector Space Model is a good fit for the concept of grammaticality grading which it modelled. The developed model is therefore suitable for grammaticality grading in natural language texts. This developed model would readily find use in computer aided language learning and automated essay scoring.

Although English language was adopted in the current study, the developed Mosesean Vector Space Model is not language dependent. Future work should be targeted at applying the developed model in evaluating the grammaticality of texts in other natural languages.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Aregbesola MK, Ganiyu RA, Olabiyisi SO, Omidiora EO, Agbele KK, Ogundokun OR. Developing grammar-checker systems using language dictionary and deep parsing. Manuscript Submitted for Publication; 2016.

[2] Aregbesola MK, Ganiyu RA, Olabiyisi SO, Omidiora EO, Alo OO. Automated grammaticality evaluation: An exploratory study of approaches. Manuscript Submitted for Publication; 2016.

[3] Kambites M. Vectors and matrices. Lecture Notes, School of Mathematics, University of Manchester; 2014.
(Retrieved 05/01/2015)
Available:http://www.maths.manchester.ac.uk/~mkambites/teaching/cs/lectures.pdf

[4] Kiela D, Clark S. A systematic study of semantic vector space model parameters. Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) @ EACL. Gothenburg, Sweden, April 26-30. 2014;21–30.

[5] Nowak R. Review of linear algebra. Lecture Note: ECE 830 Fall 2010 Statistical Signal Processing. Lecture 3; 2010.

[6] Kondratieva M. "Linear vector spaces and subspaces". Memorial University; 2008.
(Retrieved 06/01/2015)
Available:http://www.math.mun.ca/~mkondra/linalg2/la2lec1.pdf

[7] Lindeman RW. Computer graphics: Points, scalars and vectors. Lecture Notes, Department of Computer Science, Worcester Polytechnic Institute; 2008.

[8] Zybura K. Linear vector space; 2004.
(Retrieved online on 06/01/2015)
Available:http://pirate.shu.edu/~wachsmut/Teaching/MATH3912/Projects/papers/zybura_vector_spaces.pdf

[9] Luenberger DG. Optimization by vector space methods. Stanford University, Stanford, California; 1969.
(Retrieved on 06/01/2015)
Available:http://math.oregonstate.edu/~show/old/142_Luenberger.pdf, John Wiley & Sons, Inc

[10] Van Hassel R. Own lecture notes functional analysis; 2009.
(Retrieved 06/01/2015)
Available:http://www.win.tue.nl/~rvhassel/Onderwijs/2DE08-0809/OwnLectureNotesFA.pdf

[11] Bright M, Krammer D. Linear algebra lecture notes. Warwick; 2011.

[12] Zwiebach B. Linear algebra: Vector spaces and operators. MIT Open Course Ware; 2013. Available:http://ocw.mit.edu

[13] Pennington J, Socher R, Manning CD. GloVe: Global Vectorsfor Word Representation. Computer Science Department, Stanford University, Stanford, CA 94305; 2014.

[14] Schmid H. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In Proceedings of the 18th International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland. COLING. 2004;162–168.

[15] Stolcke A. Syntactic category formation with vector space grammars. In Proceedings of 13th Annual Conference of Cognitive Science Society, Chicago; 1991.

[16] Krishnamurthy J, Mitchell TM. Vector space semantic parsing: A framework for compositional vector space models. In Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, Sofia, Bulgaria. 2013;1-10.

[17] Turney PD, Pantel P. From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research. 2010;37:1.

[18] Coecke B, Grefenstette E, Sadrzadeh M. Lambek vs. Lambek: Functorial vector space semantics and string diagrams for lambek calculus. Department of Computer Science, University of Oxford; 2013.

[19] Socher R, Pennington J, Huang EH, Andrew Y. Ng, Manning CD. Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2011.

[20] Socher R, Huval B, Manning CD, Ng AY. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2012.

[21] Turney PD. Similarity of semantic relations. Computational Linguistics. 2006;32:3.

[22] Rapp R. Word sense discovery based on sense descriptor dissimilarity. In Proceedings of the Ninth Machine Translation Summit; 2003.

[23] Johnson M. Attribute-value logic and the theory of grammar. Stanford CLSI (CSLI Lecture Notes 16); 1988.

[24] Kay M. Functional grammar. In: Proceedings of the fifth Annual Meeting of the Berkeley Linguistics Society. Berkeley, California; 1979.

[25] Lau JH, Clark A, Lappin S. Measuring gradience in speakers' grammaticality judgements. In Proceedings of the 36th Annual Conference of the Cognitive Science Society (CogSci 2014), Quebec City, Canada. 2014;821-826.

[26] Aregbesola MK, Ganiyu RA, Olabiyisi SO, Omidiora EO, Alo OO. A new vector space model for grammaticality grading in handcrafted grammar. Proceedings of the 12th International Conference of the Nigeria Computer Society (NCS), Jojein Hotel and Resort, Akure, Ondo State. July 22nd – 24th. 2015;26:39-52.

[27] MacKinlay A. Pushing the boundaries of deep parsing. PhD Thesis submitted to the Department of Computing and Information Systems, The University of Melbourne, Melbourne, Australia. 2012;45-89.

[28] Aregbesola MK, Ganiyu RA, Olabiyisi SO, Omidiora EO. Development of an automated system for grammaticality evaluation using constraints-based approach. Manuscript Submitted for Publication; 2016.

[29] The British National Corpus, B. X. Oxford University Computing Services. Resource distributed on behalf of the BNC Consortium; 2007.
Available:http://www.natcorp.ox.ac.uk/

[30] The BNC Baby. Oxford University Computing Services. Resource distributed on behalf of the BNC Consortium. 2005;2.
Available:http://www.natcorp.ox.ac.uk/

[31] Foster J. Good reasons for noting bad grammar: Empirical investigations into the parsing of ungrammatical written English. Ph.D. thesis, Department of Computer Science, Trinity College, University of Dublin; 2005.

[32] Foster J. Treebanks gone bad: Parser evaluation and retraining using a treebank of ungrammatical sentences. International Journal of Document Analysis and Recognition. 2007;10(3–4):129–145.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/20090