EAI Endorsed Transactions
on Scalable Information Systems

Research Article **EAI.EU**

Sentence Reduction for Syntactic Analysis of Compound Sentences in Punjabi Language

S.K Sharma

DAV University, Jalandhar Punjab (India)

Abstract

Objective: This research paper is an attempt to develop a syntactic analysis system for compound sentences of Punjabi language. **Methods/Statistical Analysis:** Sentence simplification approach has been used for splitting compound sentences into simple sentences and then analyzing these simple sentences for syntactical error. A full form lexicon based morph, HMM based POS tagger and set of rules have been used for identification of grammatical mistakes. **Findings:** On testing an overall precision as 93.30, recall as 97.32 and F-measure as 95.25 is reported by the system. **Application/Improvements:** the system shows better performance on comparing it with existing Punjabi grammar (Precision=76.79, Recall=87.08) and Myanmar grammar checker that works on compound sentence and shows precision 83.75%.

Keywords: Grammar Checker, Compound Sentences, Sentence simplification, Punjabi Sentences.

Received on 15 October 2018, accepted on 11 January 2019, published on 30 January 2019

doi: 10.4108/eai.13-7-2018.156440

Corresponding author. Email:sanju3916@rediffmail.com

1. Introduction

Syntactic analyzer is an automated tool used in most of the natural language engineering resources. It is usually incorporated with most of the word processing systems. A lot of work has been done in grammatical error detection and correction, mainly in English language. Very little work has been done for Indian languages. Probably, Bangla grammar checker [Alam, M. Jahangir et.al (2006)] [1], Urdu [Kabir, H. et.al (2002)] [30] and Punjabi grammar checker [Singh M et.al (2008), Sharma S. K. et.al] [8], [29] are the only systems developed for Indian languages.

As the length of a sentence increases, it becomes difficult to syntactically analyze it. Therefore it becomes necessary to decrease the length of the sentence by splitting it into more than one sentences. Long sentence generally falls into category of compound sentences and complex sentences. Compound sentences are generally composed of two or more independent clauses and therefore can be split into more than one simple sentences depending upon the number of independent clauses present in the compound sentence. In most of the cases while performing the syntactic analysis on compound sentences, it becomes difficult to identify the boundary of independent clauses and hence syntactic analyzer may raise false alarm for such sentences. Therefore, it becomes essential to simplify the large sentence by reducing the length of the sentence.

1.1 Introduction to Punjabi language

Punjabi language belongs to Indo-Aryan family of languages (Indic languages). Other members that belong to this family are Hindi, Bengali, Gujarati, Marathi etc. Punjabi is spoken in India, Pakistan, Canada, USA, UK, and other countries with Punjabi immigrants. Punjabi language is the 10th most widely spoken language in the world, 4th most spoken language in Canada (The Times of India, 14th February, 2008) and the 11th in India with more than 29 million speakers. It is the official language of Punjab state. Punjabi is written in 'Gurmukhi' script in eastern Punjab (India), and in 'Shahmukhi' script in western Punjab (Pakistan). As compared to other languages like English, Punjabi is a morphologically rich language and has relatively free word order. It follows a Subject-Object-Verb (S-O-V) pattern.

2. Introduction to Machine learning

Machine learning can be defined as the field that provide the computer the capability to learn without being programmed. Sometimes it is called branch of Artificial Intelligence and generally both terms are used interchangeably. Machine learning has application in many areas including medical science [59], finance [60], query optimization, pattern recognition, healthcare sectors [61] [62] etc. It also plays an important role in Natural language processing especially in text processing, for identification of Part of speech tags, sentiment analysis, identification of entities etc. In NLP the ML technique can be expressed as model that can be used to generate other text from a given text i.e. paraphrasing [63]. It can also be used as set of instructions (algorithm) that can be used to extract useful information from the text like summarization system. Syntactic analysis of text is also one of the application of the ML.

3. Approaches Used For Syntactic Analysis

There are basically three approached used for grammar checking. These includes rule based approach [Daniel Naber, 2003] [6], syntax based approach [Jensen et al, 1993] and statistics based approach [Attwell, 1997]. The rule based approach has been used for Dutch language [Vosse, 1992][17], Czech and Bulgarian language [Kuboň and Plátek, 1994][18], English language [Adriaens, 1994][19], Swedish language[Hein, 1998][20], German language [Schmidt-Wigger, 1998][21], English language [Ravin, 1998][22], Korean language [Young-Soog, 1998][23], Danish language [Paggio, 2000][24], French, German, and Spanish languages [Helfrich and Music, 2000][25], French language [Vandeventer, 2001][26], Swedish language [Carlberger et al., 2002, 2004][27], German language [Fliedner, 2002][28], Swedish languge [Kann 2002 and Bigert et al. 2004][13], Urdu language [Kabir et al., 2002][30], English languge [Naber, 2003][6], Swedish languge [Hashemi, 2003][31], English language [Moré et al. 2004], [Rider, 2005][32], Brazilian Portuguese language [Kinoshita et al., 2006], Nepali language [Bal and Shrestha, 2007], Persian language [Ehsan and Faili, 2010][14], Afan Oromo (language widely spoken and used in Ethiopia) [Tesfaye, 2011][33], Chinese language [Jiang et al., 2011][34], Malay language [Kasbon et al., 2011][35], Tagalog Filipino (official language of the Philippines) [Oco and Borra , 2011]. Statistics based approach has been used for English language [Park et al., 1997][9], French writers writing in English [Tschichold et al., 1997][10], English language [Powers, 1997][11], Swedish [Arppe, 1999][12], Bangla and English languages [Alam et al., 2006][1], Swedish language [Sjöbergh, 2006][13], Persian language [Ehsan and Faili, 2010][14], Amharic language [Temesgen and Assabie, 2012][15], A Language Independent Statistical Grammar (LISG) checking system [Verena Henrich and Timo Reuter, 2009][16]. All these three approaches have their own advantages and drawbacks.

4. Approaches Used For Sentence Simplifications

Various techniques have been used for simplification of large sentences by different researchers. These includes Lexical Approach used for development of PSET (The Use of a Psycholinguistic Database in the Simplification of Text for Aphasic Readers) [38], KURA (Text Simplification for Reading Assistance) [44], HAPPI (Helping Aphasic People Process Online Information) [45], SIMPLEX (Putting It Simply: a Context-aware Approach to Lexical Simplification) [39], LexSiS: Lexical Simplification for Spanish [46] etc. Generation approach used for automatic Induction of rules for text simplification [36], simplification of Newspaper text to assist Aphasic reader [47], text simplification for information seeking applications [37], maintaining discourse when performing syntactic simplification [40], splitting of long sentence after explanation generation [41], developing an authoring tool which provides text simplification techniques whilst writing a document [43], acquisition of syntactic simplification rules for French [48], splitting of Vietnamese sentences for Vietnamese English machine translation [49], automatic simplification of Bosque complex sentences using dependency tree [50], sentence simplification to enhance multi-document summaries [42], development of sentence simplification tool for children's stories in Italian [51], simplification of Korean sentences for deaf readers [52], direct manipulation of parse tree [53], removing unnecessary parts of sentences [54] and Spanish sentence simplification [55].

5. Proposed model

Compound sentences are composed of at least two independent clauses joined by coordinate conjunctions, comma or semicolon and these exists in fixed patterns [2], [58]. For grammar checking of compound sentences, each clause is extracted from the sentence and grammar checking is performed on it. Since there may be two to any number of independent clauses present in compound sentences, therefore, researcher proposed divide and conquer model that can be used for grammar checking of compound sentences. In accordance with divide and conquer, the compound sentence is simplified by splitting it into individual clauses and then each individual clause undergoes error detection and correction mechanism. In this way, overall grammar checking process for compound sentences takes place in two steps; first step is to split the compound sentence into simple sentences and second step is to perform grammar checking on each simple sentence. For extracting the independent clauses from the compound sentence, the clause boundary of these clauses is identified in the similar way as in [56] and [57]. Researcher's system checks the errors in the sentence at phrase level, clause level and then at sentence level. For phrase level and clause level, rule based approach has been followed. For sentence level, this rule based approach has been extended to all the clauses of the sentence. The syntactic analysis in compound sentences takes place in three phases:

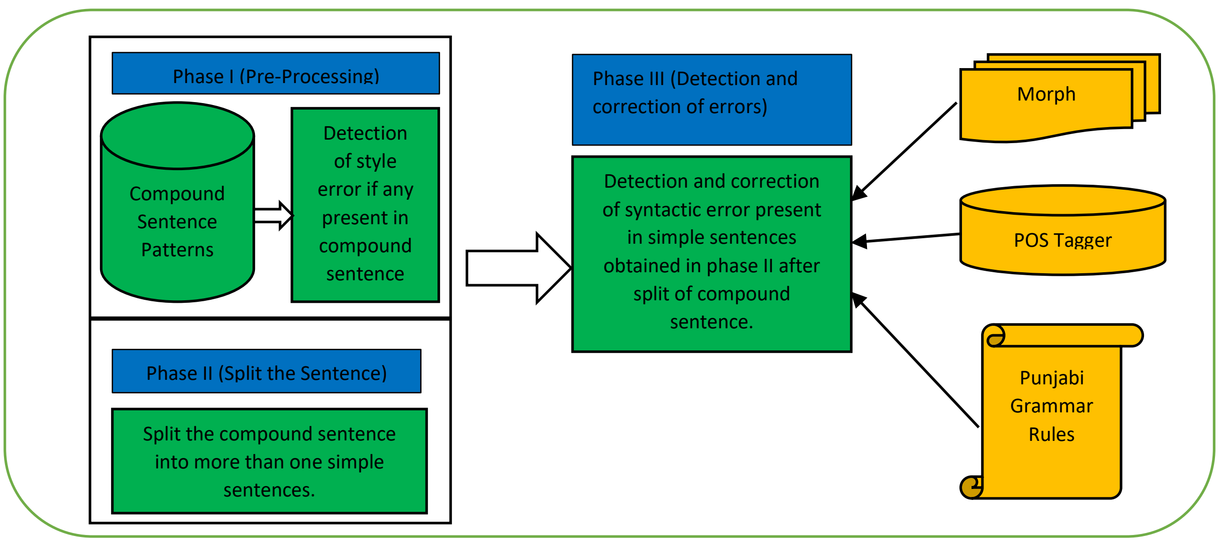

Figure 1: Proposed model for syntactic analysis of compound sentences in Punjabi language

5.1 Phase I

In this phase, style error at the sentence level is identified and rectified. This is done by analyzing the complete structure of the input compound sentence. Compound sentences have fixed patterns of structure [2], [3]. An input compound sentence is matched against these fixed patterns. In order to match the input sentence against these patterns, a database containing all the possible patterns of compound sentences has been developed and stored in the form of regular expression.

5.2 Phase II

In the second phase, internal structure of compound sentences is analyzed for detection of errors. In the internal structure, errors at phrase level and then at clause level are detected and rectified. In order to perform error detection and correction at phrase and clause levels, each input compound sentence is simplified by splitting the sentence into independent clauses (simple sentences). This simplification of sentences is performed on the basis of clause boundary mark information [56], [57].

5.3 Phase III

To check the grammatical errors in simple sentences or in independent clauses of Punjabi sentences, 'government and binding' prevalent in Punjabi sentences has been studied. As per 'government and binding' prevalent, there exists a grammatical agreement between various components of a sentence like modifier and noun agreement, subject/object and verb agreement, noun and adjective agreement, noun phrase in oblique form before postposition etc. Also, as per government and binding, all the words present in an independent clause must grammatically agree with the head word of that clause. The head word of the clause is the first phrase head that is present in the noun phrase of the clause. Various types of errors detected and corrected by the system developed by the researcher has been listed in table 1. For each error type mentioned in table 1, a separate module has been developed. Each module detects and rectifies a specific type of error. During detection of error, all these modules are executed in sequence.

The algorithm used in the grammar checking of compound sentences is as following:

Algorithm: Error detection in compound sentences

Databases used: Error type.

Input: Incorrect simple sentence (independent clause obtained after simplification)

Output: Corrected simple sentence.

1. Get all the error type that have *OnOff* value set to 1, from the error type database sorted by the *Priority* field
2. Repeat steps 3 to 5 for current clause.
3. Repeat steps 4 and 5 for all the error types.
4. Call the respective module to perform the correction on the current clause.
5. Output the corrected sentence.

Consider the following incorrect compound sentence:

Incorrect sentence:

Punjabi: ਦੋ ਮੋਟਾ ਮੁੰਡੇ ਭੱਜ ਰਿਹਾ ਸੀ ਅਤੇ ਪੁਲਿਸ ਓਹਨਾਂ ਦਾ ਪਿੱਛਾ ਕਰ ਰਿਹਾ ਸੀ।

Roman Transliteration: (dō mōṭā muṇḍē bhajj rihā sī atē pulis ōhnāṃ dā picchā kar rihā sī.)

English translation: (Two fatty boys run+ing was and cop them chase + ing was)
Two fatty boys was running and cop was chasing them

In above incorrect sentence, there are two clauses in the compound sentence and each clause contains errors. The first clause is "ਦੋ ਮੋਟਾ ਮੁੰਡੇ ਭੱਜ ਰਿਹਾ ਸੀ " (dō mōṭā muṇḍē bhajj rihā sī) and it contains noun modifier agreement error as the modifier ਮੋਟਾ (mōṭā) (singular) does not grammatically agree with noun ਮੁੰਡੇ (muṇḍē) (plural) in terms of number. Second clause is ਪੁਲਿਸ ਓਹਨਾਂ ਦਾ ਪਿੱਛਾ ਕਰ ਰਿਹਾ ਸੀ (pulis ōhnāṃ dā picchā kar rihā sī.), and it contains subject verb agreement error as the object ਪੁਲਿਸ (pulis) (feminine) does not grammatically agree with verb ਰਿਹਾ (rihā) (masculine) in terms of gender. Both these errors are detected and rectified in two steps. In the first step, two clauses are separated from the sentence and in the second step, these clauses are detected for the presence of error. After applying detection and correction on individual clauses, the final updated output given by the researcher's system is:

Corrected sentence:

Punjabi: ਦੋ ਮੋਟੇ ਮੁੰਡੇ ਭੱਜ ਰਹੇ ਸੀ ਅਤੇ ਪੁਲਿਸ ਓਹਨਾਂ ਦਾ ਪਿੱਛਾ ਕਰ ਰਹੀ ਸੀ ।

Roman Transliteration: (dō mōṭē muṇḍē bhajj rahē sī atē pulis ōhnāṃ dā picchā kar rahī sī .)

English Translation: (Two fatty boys run+ing were and cop them chase + ing was)
Two fatty boys were running and cop was chasing them

The complete architecture of the above method with example has been shown in figure 2. It is clear from figure 2 that the compound sentence is first simplified by splitting it at the conjunction and separating each independent clause. Then each independent clause is passed through grammar checking system where error detection and correction algorithm mention above is applied.

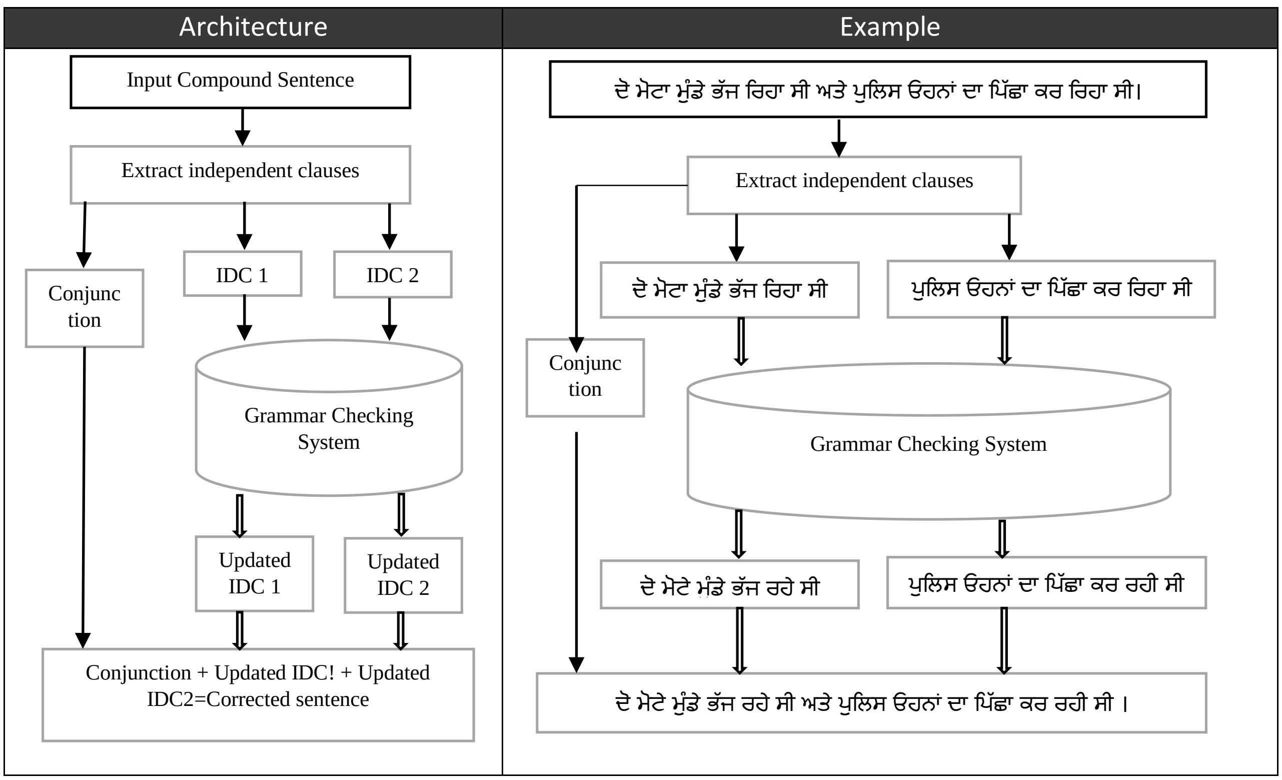

Figure 2: Architecture of grammar checking of compound sentence

6. Error Covered

Various types of syntactic errors in an independent clause handled by researcher's system are listed in table 1. These errors are basically categorized into three classes; first is agreement error; second is postposition related errors and third is error due to order of words in noun and verb phrases. First column of the table represents the error category, second column represents the description of the grammatical mistake and third column shows the example containing incorrect and correct sentences related with the corresponding error shown in second column.

Table 1: Various Error types handled by the system

Error Category	Description of error	Examples
Agreement error	All the Noun phrases joined by conjunctions to form group must agree in case.	Incorrect: ਮੁੰਡਾ ਅਤੇ ਕੁੜੀਆਂ ਨੇ ਸਮਾਗਮ ਵਿੱਚ ਵੱਧ ਚੜ੍ਹ ਕੇ ਹਿੱਸਾ ਲਿਆ। (muṇḍā atē kuḍaīāṃ nē samāgam vicc vaddh caḍah kē hissā liā .) Correct: ਮੁੰਡੇ ਅਤੇ ਕੁੜੀਆਂ ਨੇ ਸਮਾਗਮ ਵਿੱਚ ਵੱਧ ਚੜ੍ਹ ਕੇ ਹਿੱਸਾ ਲਿਆ। (muṇḍē atē kuḍaīāṃ nē samāgam vicc vaddh caḍah kē hissā liā .)
	Noun Modifier Agreement	Incorrect: ਕਾਲਾ ਮੁੰਡੇ ਖੇਡ ਰਹੇ ਸੀ। (kālā muṇḍē khēḍ rahē sī .) Correct: ਕਾਲੇ ਮੁੰਡੇ ਖੇਡ ਰਹੇ ਸੀ। (kālē muṇḍē khēḍ rahē sī .)
	Adjective Phrase Agreement	Incorrect: ਤੇਰਾ ਪਿਨ ਸੋਹਣੀ ਲਿਖਦੀ ਹੈ। (tērā pin sōhṇī likhdī hai.) Correct: ਤੇਰਾ ਪਿਨ ਸੋਹਣਾ ਲਿਖਦਾ ਹੈ। (tērā pin sōhṇā likhdā hai.)
	Subject Verb Agreement	Incorrect: ਮੁੰਡਾ ਖੇਡ ਰਹੀ ਹੁੰਦੀ ਹੈ। (muṇḍā khēḍ rahī hundī hai.) Correct: ਮੁੰਡਾ ਖੇਡ ਰਿਹਾ ਹੁੰਦਾ ਹੈ। (muṇḍā khēḍ rihā hundā hai.)
	In a phrase or clause, all the words in agreement with headword must have same gender and number.	Incorrect:ਇਹ ਮੇਰਾ ਘਰ ਹਨ। (ih mērā ghar han.) Correct: ਇਹ ਮੇਰਾ ਘਰ ਹੈ। (ih mērā ghar hai.)
Postposition related error	Noun phrase must be in oblique form before postposition.	Incorrect:ਮੁੰਡਾ ਨੂੰ ਜਾਣਾ ਪੈ ਗਿਆ।(muṇḍā nūṃ jāṇā pai giā.) Correct: ਮੁੰਡੇ ਨੂੰ ਜਾਣਾ ਪੈ ਗਿਆ। (muṇḍē nūṃ jāṇā pai giā.)
	DAA (ਦਾ) postposition should be in agreement with noun phrase in terms of number and gender	Incorrect: ਮੁੰਡੇ ਦਾ ਕਿਤਾਬ ਗੁੰਮ ਹੋ ਗਈ। (muṇḍē dā kitāb gumm hō gaī.) Correct: ਮੁੰਡੇ ਦੀ ਕਿਤਾਬ ਗੁੰਮ ਹੋ ਗਈ। (muṇḍē dī kitāb gumm hō gaī.)
Order of words in a phrase related error	Order of modifier in Noun phrase	Incorrect: ਸੋਹਣੇ ਪੰਜਵੇਂ ਮੁੰਡੇ ਨੇ ਕਿਹਾ। (sōhṇē pañjvēṃ muṇḍē nē kihā.) Correct: ਪੰਜਵੇਂ ਸੋਹਣੇ ਮੁੰਡੇ ਨੇ ਕਿਹਾ। (pañjvēṃ sōhṇē muṇḍē nē kihā.)
	Order of words in a verb phrase	Incorrect:ਮੁੰਡਾ ਆਇਆ ਨਹੀਂ ਹੋਵੇਗੀ। (muṇḍā āiā nahīṃ hōvēgī.) Correct: ਮੁੰਡਾ ਆਇਆ ਨਹੀਂ ਹੋਵੇਗਾ। (muṇḍā āiā nahīṃ hōvēgā.)

Each independent clause is checked for various grammatical errors. All these errors are detected in sequence as mention in table 1.

7. Result and discussion:

For testing the final module of syntactic analyzer for compound sentences, researcher first input the dummy test data. This dummy data contains the sentences having the errors for which the system is developed. After testing with dummy test data, real test data (data collected from hand written test papers by 6th to 10th standard students studying Punjabi as second language) and Hindi to Punjabi machine translated test data (http://h2p.learnpunjabi.org/) has been used to test the system. Number of sentences used from each type of data is mentioned in table 2. The results obtained have been tabulated in table 3.1, 3.2, 3.3, and figure 3.1, 3.2, 3.3.

Table 2: Test data used to test various types of error

Sr. No.	Main error		Sub-category of main error	Number of sentences from dummy test data	Number of sentences from real data	Number of sentences from Hindi to Punjabi Machine translation system
1.	Modifier and Noun agreement	1.1	Agreement in terms of Number	220	18	45
		1.2	Agreement in terms of Gender	200	23	58
		1.3	Agreement in terms of Case	45	12	30
2.	Subject Verb agreement	2.1	Agreement in terms of Number	200	13	30
		2.2	Agreement in terms of Gender	156	12	30
		2.3	Agreement in terms of Person	223	13	33
		2.4	Use of postposition with subject if the verb is transitive in perfect form	96	8	20
3.	Noun and Adjective agreement	3.1	Agreement in terms of Number	269	13	35
		3.2	Agreement in terms of Gender	210	23	54
		3.3	Agreement in terms of Case	222	14	35
4.	Order of modifier of Noun phrase	4.1	Pronoun precedes all other modifiers	104	23	55
		4.2	Numeral precedes adjective	86	4	10
5.	Order of word in word phrase			537	49	121
6.	Agreement of noun phrase with DA postposition			29	32	80
7.	Oblique case of noun before Postposition			132	12	30
Total Number of sentences				2729	269	666

Table 3.1: Experimental evaluation of grammar checking of compound sentences (Dummy test data)

Error type	Number of Errors in input sentences [A]	No. of errors corrected by the system [B]	No. of errors wrongly corrected by the system [C]	Recall $\frac{B+C}{A}$ X 100	Precision $\frac{B}{A}$X 100	F score $\frac{Precision\ X\ Recall}{precision+recall}$X2
Modifier and Noun Agreement (MNA)	465	463	0	99.56989	99.56989	99.56989
Subject Verb Agreement (SVA)	675	674	1	100	99.85185	99.92587
Noun and Adjective Agreement (NAA)	701	701	0	100	100	100
Order of Modifier of a Noun Phrase (OMNP)	190	187	2	99.47368	98.42105	98.94457
Order of Words in a Verb Phrase (OWVP)	537	531	5	99.81378	98.88268	99.34605
Due to Postposition DA (PDA)	29	23	4	93.10345	79.31034	85.65517
Due to Postposition NU (PNU)	132	128	3	99.24242	96.9697	98.0929
Overall Average	2729	2707	15	99.7435	99.19384	99.46791

Table 3.2: Experimental evaluation of grammar checking of compound sentences (Real test data)

Error type	Number of Errors in input sentences [A]	No. of errors corrected by the system [B]	No. of errors wrongly corrected by the system [C]	Recall $\frac{B+C}{A}$ X 100	Precision $\frac{B}{A}$X 100	F score $\frac{Precision\ X\ Recall}{precision+recall}$X2
Modifier and Noun Agreement (MNA)	53	49	2	96.22642	92.45283	94.30189
Subject Verb Agreement (SVA)	46	43	2	97.82609	93.47826	95.60277

Noun and Adjective Agreement (NAA)	50	44	3	94	88	90.9011
Order of Modifier of a Noun Phrase (OMNP)	27	23	3	96.2963	85.18519	90.4006
Order of Words in a Verb Phrase (OWVP)	49	42	5	95.91837	85.71429	90.5297
Due to Postposition DA (PDA)	32	28	2	93.75	87.5	90.51724
Due to Postposition NU (PNU)	12	10	1	91.66667	83.33333	87.30159
Overall Average	269	239	18	95.53903	88.84758	92.07189

Table 3.3: Experimental evaluation of grammar checking of compound sentences (Hindi to Punjabi Machine translation data)

Error type	Number of Errors in input sentences [A]	No. of errors corrected by the system [B]	No. of errors wrongly corrected by the system [C]	Recall $\frac{B+C}{A}$ X 100	Precision $\frac{B}{A}$X 100	F score $\frac{Precision\ X\ Recall}{precision+recall}$X2
Modifier and Noun Agreement (MNA)	133	125	5	97.74436	93.98496	95.8278
Subject Verb Agreement (SVA)	113	102	6	95.57522	90.26549	92.8445
Noun and Adjective Agreement (NAA)	124	118	4	98.3871	95.16129	96.74731
Order of Modifier of a Noun Phrase (OMNP)	65	58	3	93.84615	89.23077	91.48028
Order of Words in a Verb Phrase (OWVP)	121	116	3	98.34711	95.86777	97.09161
Due to Postposition DA (PDA)	80	69	6	93.75	86.25	89.84375

Due to Postposition NU (PNU)	30	24	5	96.66667	80	87.54717
Overall Average	666	612	32	96.6967	91.89189	94.23309

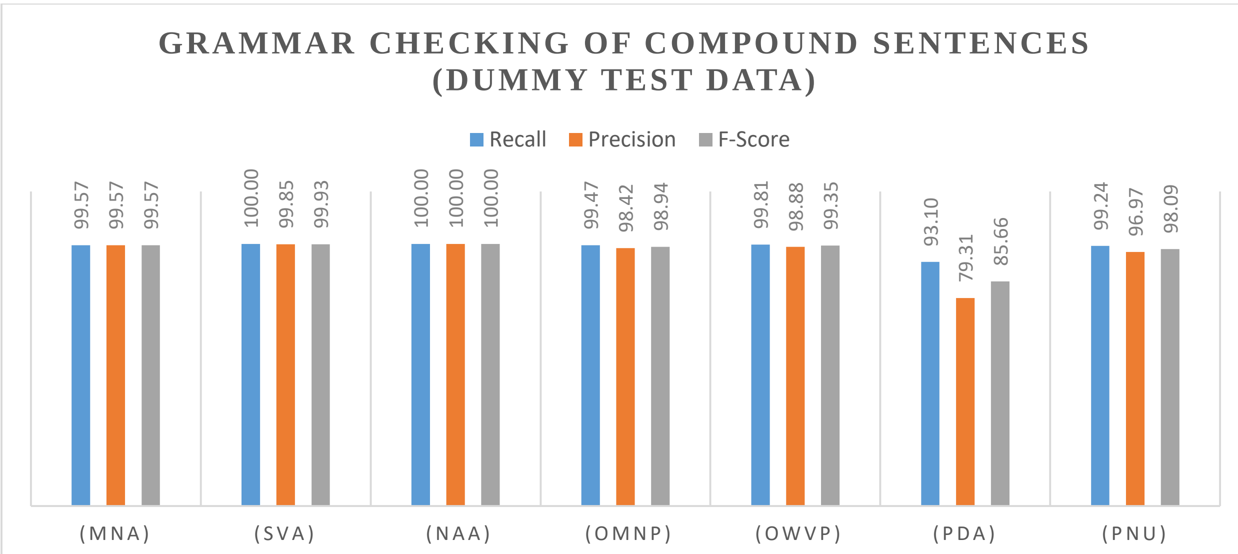

Figure 3.1: Experimental Evaluation of Grammar checking of compound sentences (Dummy Test Data)

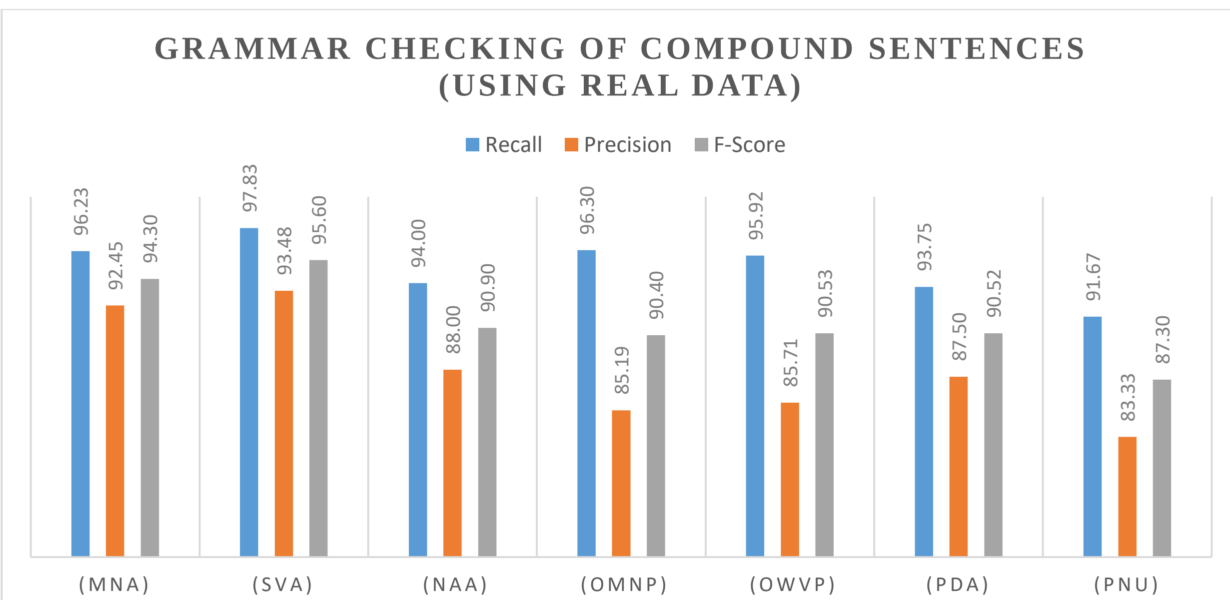

Figure 3.2: Experimental Evaluation of Grammar checking of compound sentences (Using Real Data)

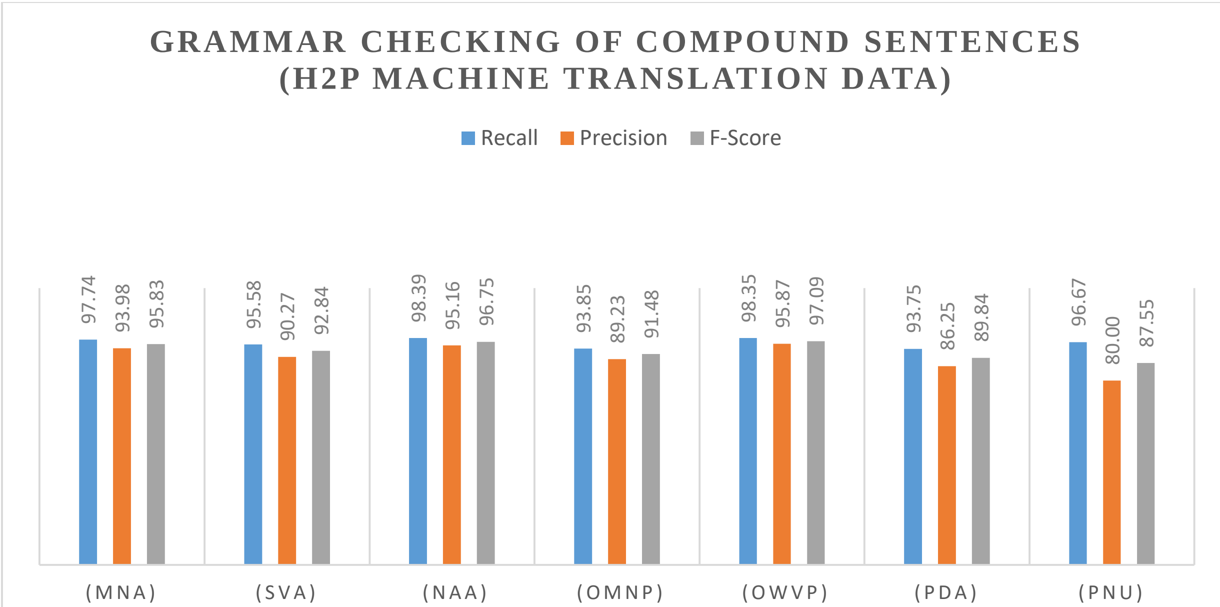

Figure 3.3: Experimental Evaluation of Grammar checking of compound sentences (Using H2P Machine Translation Data)

As shown in tables 3.1, 3.2, 3.3 and figure 3.1, 3.2 and 3.3 the developed system shows a recall as 99.74, precision as 99.14 and F-score as 99.46 for Dummy test data, recall as 95.53, precision as 88.84 and F-score as 92.07 for Real test data and recall as 96.69, precision as 91.89 and F-score as 94.23 for H2P machine translation test data. An overall precision as 93.30, recall as 93.30 and F-measure as 95.25 is reported by the system. Low precision (80%) is also reported in some cases (Error Due to postposition NU in case of H2P machine translation test data) due to presentce of unknown or misspelled words in the input sentences. Further the developed system will show low performance when there are number of spelling or typing mistakes in the Punjabi words. Also the system will not work in case of compound – complex sentences because the complex sentences can not be easily split into dependent and independent clauses.

8. Comparison with the existing system

This system can be compared only partially with the existing system because the existing grammar checker for Punjabi language has been developed to check the grammar mainly for simple sentences whereas researcher's proposed system mainly deals with compound sentences. The existing Punjabi grammar checker system [8] reports precision of 76.79%, recall of 87.08%, and F-measure of 81.61%. Researcher's system is tested for compound sentences for evaluating the results. For compound sentences, this system shows an overall precision as 93.30, recall as 97.32 and F-measure as 95.25. Also the developed system cannot be compared with the existing systems developed for other languages, due to the coverage of different types of errors by different grammar checking systems. As per reviewed literature the only grammar checking system that deal with compound sentences is the system developed by Lin et al. (2011) [58] for Myanmar language and this system shows a precision of 83.75% for compound sentences.

9. Conclusion and future scope

In this research author has developed a syntactic analyzer system for compound sentences of Punjabi language that shows an overall precision as 93.30, recall as 97.32 and F-measure as 95.25. Compound sentence simplification approach used in this work can be further used for generating paraphrasing of Punjabi language. Further this compound sentence simplification work can be extended to complex sentence simplification work. Similar grammar checker technique can be implemented to other languages that are similar to Punjabi language like Hindi. Further this work can be extended for grammar checking of complex sentences.

References

[1]. Alam, Md. Jahangir, Naushad UzZaman, and Mumit Khan. (2006). N-gram based Statistical Grammar Checker for Bangla and English. In Proc. of ninth International Conference on Computer and Information Technology (ICCIT 2006), Dhaka,Bangladesh.

[2]. Cheema B S. (2005). Punjabi vaak prabandh (banatar ate karaj) , Publication Bureau, Punjabi University, Patiala, India.

[3]. Brar, B S, (2008). Punjabi viakran (Sidhant ate vihar) , Publication Bureau, Punjabi University, Patiala, India.

[4]. Duni, C. (1964). Punjabi Bhasha da Viakaran (Punjabi). Punjab University Publication Bureau, Chandigarh, India.

[5]. Gill, H S. and Henry A. Gleason, Jr. (1986). A Reference Grammar of Punjabi. Publication Bureau, Punjabi University, Patiala, India.

[6]. D. Naber. (2003). A Rule-Based Style and Grammar Checker, Diploma Thesis, Computer Science-Applied, University of Bielefeld.

[7]. Gill, Harjeet S. and Henry A. Gleason, Jr. 1986. A Reference Grammar of Punjabi. Publication Bureau, Punjabi University, Patiala, India.

[8]. Gill M S, Lehal G S (2008) "Grammer Checking System for Punjabi" Coling 2008: companion volume Posters and Demonstrations pages 149–152 Manchester.

[9]. Park, J. C., Palmer, M. S., & Washburn, C. (1997). An English Grammar Checker as a Writing Aid for Students of English as a Second Language. In ANLP p. 24.

[10]. Tschichold, C., Bodmer, F., Cornu, E., Grosjean, F., Grosjean, L., Kübler, N. & Tschumi, C. (1997). Developing a new grammar checker for English as a second language. Proc. From Research to Commercial Applications: Making NLP Work in Practice, 7-12.

[11]. Powers, D. M. (1997). Learning and application of differential grammars. InProc. Meeting of the ACL Special Interest Group in Natural Language Learning, Madrid.

[12]. Arppe, A. (2000). Developing a grammar checker for Swedish. In The 12th Nordic Conference of Computational Linguistics pp. 13-27.

[13]. Bigert, J., Kann, V., Knutsson, O., & Sjöbergh, J. (2004). Grammar checking for Swedish second language learners. pp. 33-47.

[14]. Ehsan, N., & Faili, H. (2010). Towards grammar checker development for Persian language. IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), pp. 1-8

[15]. Temesgen, A., & Assabie, Y. (2013). Development of Amharic Grammar Checker Using Morphological Features of Words and N-Gram Based Probabilistic Methods. IWPT-2013, p. 106.

[16]. Henrich, V. (2009). LISGrammarChecker: Language Independent Statistical Grammar Checking (Doctoral dissertation, Reykjavík University).

[17]. Vosse, T. (1992). Detecting and correcting morpho-syntactic errors in real texts. In *Proceedings of the third conference on applied natural language processing*. Association for Computational Linguistics. pp. 111-118

[18]. Kuboň, V., & Plátek, M. (1994). A grammar based approach to a grammar checking of free word order languages. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. pp. 906-910

[19]. Adriaens, G. (1994). The LRE SECC Project: Simplified English Grammar and Style Correction in an MT Framework. In *Language engineering convention* pp. 1-8.

[20]. Hein, A. S. (1998). A Chart-Based Framework for Grammar Checking Initial Studies. In *Proc. of 11th Nordic Conference in Computational Linguistic*. pp. 68-80.

[21]. Schmidt-Wigger, A. (1998). Grammar and style checking for German. In Proceedings *of CLAW* (Vol. 98).

[22]. Ravin, Y. (1993). Grammar Errors and Style Weaknesses in a Text-Critiquing System. In *Natural Language Processing: The PLNLP Approach*. Springer US. pp. 65-76.

[23]. Young-Soog, C. (1998). Improvement of Korean Proofreading System Using Corpus and Collocation Rules. *Language*, pp. *328*-333.

[24]. Paggio, P. (2000). Spelling and grammar correction for Danish in SCARRIE.

In *Proceedings of the sixth conference on applied natural language processing*. Association for Computational Linguistics. pp. 255-261.

[25]. Helfrich, A., & Music, B. (2000). Design and evaluation of grammar checkers in multiple languages. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. pp. 1036-1040

[26]. Vandeventer, A. (2001). Creating a grammar checker for CALL by constraint relaxation: a feasibility study. *ReCALL*, *13*(01), pp. 110-120.

[27]. Carlberger, J., Domeij, R., Kann, V., & Knutsson, O. (2002). A Swedish grammar checker. Submitted to Comp. *Linguistics, oktober*.

[28]. Fliedner, G. (2002). A system for checking NP agreement in German texts. In *Proceedings of the ACL Student Research Workshop.* pp. 12-17.

[29]. Sharma S K, Lehal G S. (2016). Improving Existing Punjabi Grammar Checker. IEEE International Conference on Computation Techniques in Information and Communication Technologies held at Indraprastha University, New Delhi, IEEE Xplore,.

[30]. Kabir, H., Nayyer, S., Zaman, J., & Hussain, S. (2002, December). Two Pass Parsing Implementation for an Urdu Grammar Checker. In *Multi Topic Conference, 2002. Abstracts. INMIC 2002. International* (pp. 51-51). IEEE.

[31]. Hashemi, S. S. (2007). Ambiguity resolution by reordering rules in text containing errors. In *Proceedings of the 10th International Conference on Parsing Technologies*. Association for Computational Linguistics. pp. 69-79.

[32]. Rider, Z. (2005). Grammar checking using POS tagging and rules matching. In *Class of 2005 Senior Conference on Natural Language Processing*.

[33]. Tesfaye, D. (2011). A rule-based Afan Oromo Grammar Checker. IJACSA Editorial.

[34]. Jiang, Y., Wang, T., Lin, T., Wang, F., Cheng, W., Liu, X., & Zhang, W. (2012). A rule based Chinese spelling and grammar detection system utility. IEEE International Conference on System Science and Engineering (ICSSE), 2012. pp. 437-440.

[35]. Kasbon, R., Amran, N., Mazlan, E., & Mahamad, S. (2011). Malay language sentence checker. World Appl. Sci. J.(Special Issue on Computer Applications and Knowledge Management), 12, pp. 19-25.

[36]. Chandrasekar, R., Srinivas, B. (1997). Automatic induction of rules for text simplification. Knowl.-Based Syst. 10(3), 183–190.

[37]. Klebanov, B.B., Knight, K., Marcu, D. (2004). Text simplification for information-seeking applications. In: Meersman, R., Tari, Z. (eds.) OTM 2004. LNCS, vol. 3290, pp. 735–747. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30468-5_47

[38]. Coster, W., Kauchak, D. (2011). Learning to simplify sentences using Wikipedia. In: Proceedings of the Workshop on Monolingual Text-To-Text Generation. Association for Computational Linguistics, Portland, June 2011, pp. 1–9.

[39]. Biran, O., Brody, S., Elhadad, N.(2011). Putting it simply: a context-aware approach to lexical simplification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, vol. 2. Association for Computational Linguistics, Stroudsburg, pp. 496–501.

[40]. Siddharthan, A. (2006). Syntactic simplification and text cohesion. Res. Lang. Comput. 4, 77–109.

[41]. Kandula, S., Curtis, D., Zeng-Treitler, Q.(2010). A semantic and syntactic text simplification tool for health content. In: AMIA Annual Symposium Proceedings, pp. 366–370.

[42]. Silveira, S., Branco, A. (2012). Combining a double clustering approach with sentence simplification to produce highly informative multi- document summaries. In: 2012 IEEE 13th International Conference on Information Reuse and Integration (IRI), pp. 482–489.

[43]. Scarton, C., de Oliveira, M., Candido Jr., A., Gasperin, C., Aluísio, S.M.(2010). Simplifica: a tool for authoring simplified texts in Brazilian Portuguese guided by readability assessments. In: Proceedings of the NAACL HLT 2010 Demonstration Session, pp. 41–44.

[44]. Inui, K., Fujita, A., Takahashi, T., Iida, R., Iwakura, T.(2003). Text simplification for reading assistance: a project note. In: Proceedings of the Second International Workshop on Paraphrasing, pp. 9–16.

[45]. Devlin, S., Unthank, G.(2006). Helping aphasic people process online information. In: Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 225–226.

[46]. Bott, S., Rello, L., Drndarević, B., Saggion, H.(2012): Can Spanish be simpler? LexSiS: lexical simplification for Spanish. In: Proceedings of COLING 2012. The COLING 2012. Organizing Committee, Mumbai, India, pp. 357–374.

[47]. Carroll, J., Minnen, G., Canning, Y., Devlin, S., Tait, J. (1998). Practical simplification of English newspaper text to assist aphasic readers. In: Proceedings of AAAI-1998 Workshop on

Integrating Artificial Intelligence and Assistive Technology, pp. 7–10.
[48]. Seretan, V. (2012). Acquisition of syntactic simplification rules for French. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012). European Language Resources Association (ELRA), Istanbul, May 2012
[49]. Hung, B.T., Minh, N.L., Shimazu, A.(2012). Sentence splitting for Vietnamese-English machine translation. In: 2012 Fourth International Conference on Knowledge and Systems Engineering (KSE), pp. 156–160.
[50]. Aranzabe, M.J., de Ilarraza, A.D., Gonzalez-Dios, I. (2012). Transforming complex sentences using dependency trees for automatic text simplification in Basque. Procesamiento del Lenguaje Natural 50, 61–68.
[51]. Barlacchi, G., Tonelli, S.: ERNESTA (2013). A sentence simplification tool for children’s stories in Italian. In: Gelbukh, A. (ed.) CICLing 2013. LNCS, vol. 7817, pp. 476–487. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37256-8_39
[52]. Chung, J.-W., Min, H.-J., Kim, J., Park, J.C. (2013). Enhancing readability of web documents by text augmentation for deaf people. In: Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, pp. 30:1–30:10. ACM, New York (2013)
[53]. Feblowitz, D., Kauchak, D. (2013). Sentence simplification as tree transduction. In: Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations, pp. 1–10. Association for Computational Linguistics, Sofia, August 2013
[54]. Klerke, S., Søgaard, A.(2013). Simple, readable sub-sentences. In: 51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop, pp. 142–149. Association for Computational Linguistics, Sofia, August 2013
[55]. Štajner, S., Drndarević, B., Saggion, H.(2013). Corpus-based sentence deletion and split decisions for Spanish text simplification. Revista Computación y Sistemas 17(2).
[56]. Sharma, S. K. (2017). Marking Clause Boundaries in Compound Sentences of Punjabi Language. International Journal of Computer Science and Engineering, 84-88, 5(9).
[57]. Sharma S. K, Lehal G S. (2015) Identification of clause boundary in Punjabi language. International conference on advanced in Computer, Communication and Electronic Engineering held at University of Kashmir.
[58]. Lin, N. Y., Soe, K. M., & Thein, N. L. (2011). Developing a chunk-based grammar checker for translated English sentences. In Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation.
[59]. Kaur, P., Sharma, M., & Mittal, M. (2018). Big Data and Machine Learning Based Secure Healthcare Framework. Procedia Computer Science, 132, 1049-1059.
[60]. Sharma, M., Sharma, S., & Singh, G. (2018). Performance Analysis of Statistical and Supervised Learning Techniques in Stock Data Mining. Data, 3(4), 54.
[61]. Sharma, M., Singh, G., & Singh, R. (2018). An Advanced Conceptual Diagnostic Healthcare Framework for Diabetes and Cardiovascular Disorders. EAI ENDORSED TRANSACTIONS ON SCALABLE INFORMATION SYSTEMS, 5(18).
[62]. Sharma, M., G. Singh, and R. Singh. (2017) "Stark Assessment of Lifestyle Based Human Disorders Using Data Mining Based Learning Techniques." IRBM.
[63]. Kozareva, Z., & Montoyo, A. (2006). Paraphrase identification on the basis of supervised machine learning techniques. In Advances in natural language processing (pp. 524-533). Springer, Berlin, Heidelberg.