Improving Generation in Machine Translation by Separating Syntactic and Morphological Processes

Nayyara Karamat
Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan
nayyara.karamat@kics.edu.pk

Kamran Malik
Punjab University College of Information Technology
Lahore, Pakistan
mkamranmalik@gmail.com

Sarmad Hussain
Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan
sarmad.hussain@kics.edu.pk

Abstract-**This paper presents a generation approach in a Lexical Functional Grammar (LFG) based machine translation system that subdivides the process and uses rule based modules to address the problem. The results show improvement in performance compared to the earlier work which generates the translation into Urdu using a single integrated process.**

I. Introduction

This paper presents continuation of the work on English to Urdu machine translation (MT) reported earlier [1]. The architecture of the MT system is divided into three traditional components, a Parser, a Mapper and a Generator. Figure 1 shows these components.

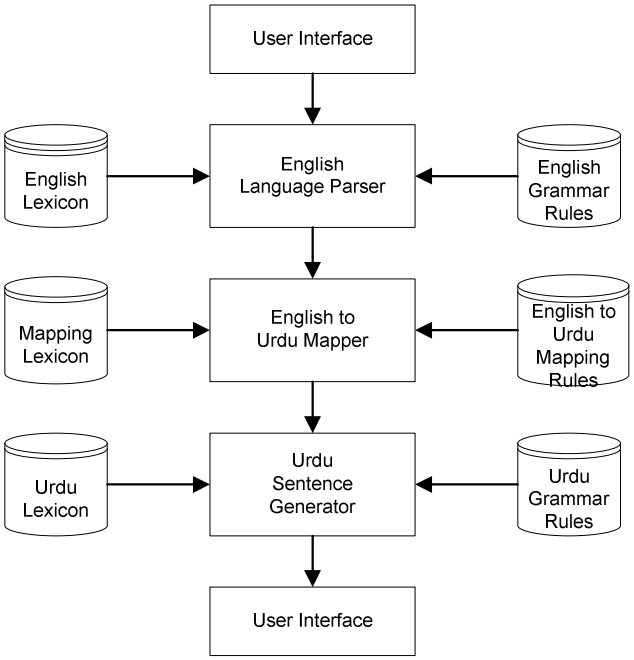

Figure 1: English to Urdu MT Architecture

The system is based on Lexical Functional Grammar (LFG), which is a unification-based linguistic formalism which is suitable for computational purposes. LFG uses different structures for representing various levels of linguistic information in a sentence, for example, constituent structure (c-structure) and functional structure (f-structure) [2]. In the system, English sentences are parsed with an LFG parser using a hand written English grammar and lexicon generating an English constituent structure (c-structure) and functional structure (f-structure) [2]. The f-structure is assumed to be relatively language independent information and is passed to the Mapper, whereas the constituent structure (c-structure) is discarded [2]. Using the English f-structure and translating all the predicates, the Mapper generates a new f-structure for Urdu. Some language specific transformations are also made during the mapping process [3]. A mapping grammar and lexicon are used for this purpose. The resultant Urdu f-structure is passed to the Generator which generates an Urdu sentence according to the given Urdu grammar and lexicon.

The English lexicon contains 10,000 root words, whereas the mapping lexicon and Urdu lexicon contain the corresponding translations in Urdu. Longman dictionary of contemporary English [4] is used to list down the senses of English words and its descriptions are used as guide in translating the words. English grammar analysis is mainly guided by Quirk et al. [5]. A number of Urdu grammar books are also consulted for developing the Urdu grammar [6,7,8,9,10,11,12]. A morphological generator is used to generate all word forms for English and Urdu lexicons offline. The MT system, grammars and their analysis documents have been released[1].

This system works well for small grammatically correct sentences (e.g. sentences up to 10 words). However, when it comes to more complex sentences, e.g., in online text with sentences having as many as 20-30 words, the system shows very slow response or is unable to translate due to the increased complexity [13].

The system was re-analyzed for the reasons and issues have been identified in the parsing and generating processes. Another issue has been the coverage of grammars. A ma-

[1] Urdu Lexicon Project, Ministry of IT, Govt. of Pakistan, http://cle.org.pk/software/langproc/E2UMachineTranslationSystem.htm and http://cle.org.pk/software/ling_resources.htm

nually developed grammar has been used in the Parser which requires time and effort involving linguistic expertise. Specifying F-description annotation adds more complexity to grammar development. Thus, there remain some linguistic phenomena which are not handled in the grammar which cause parsing failure. Similar issues exist with the Generator grammar. Urdu being a free word order and morphologically rich language further increases the complexity of the grammar and morphological constraints. The Generator is been designed in a way that if some f-structure nodes are not handled in the grammar, they are skipped in the translation altogether, though giving results, but still incomplete translations.

Further, due to the large size of the grammar, the performance is not efficient. As the grammar rules increase, computational effort for parsing and generation also increases, thereby affecting the efficiency of the system.

The architectures of both the parser and the generator have been revised to address these issues. The revised hybrid model of the parser is discussed in [14] and [15]. This paper discusses the revised generation module in more detail. In this approach, the generation problem is divided in two sub-tasks to reduce the complexity of the process. The first task is the generation of morphologically relevant from of words and the second is the sequencing of these words in the correct order.

Section 2 describes background and key ideas used in this work. In Section 3, the architecture of the generation system is discussed. In Section 4 the test results are presented. Conclusions are presented in Section 5.

II. Literature Review

Whitelock [16] introduces a "Shake and Bake" approach for generation. In this approach [17, 18, 19], target language signs are combined to form a valid sentence after trying all permutations and picking the valid one according to the target language grammar. This approach is of exponential complexity even using a chart. Many techniques are employed to improve the generation efficiency. For example, [18] employs a chart to avoid recalculating the same combinations of signs more than once during testing, and [20] proposes a technique for storing the rules which have been attempted; another technique avoids certain cases by employing global constraints on the solution space [21]; other works, such as [22] and [23], provide a system for bag generation that is heuristically guided by probabilities, and [24] presents a polynomial time algorithm for generation by restricting the target grammar and using restrictive data structure instead of bag of signs.

The idea of dividing generation problem into sub problems and solving them separately is also used in Statistical Machine Translation (SMT) systems. There has been work done on handling morphological issues in target language, target language capitalization [25], case marker generation [26, 27] and inflection generation [28]. The algorithm presented in this paper also employs the idea of dividing the problem into sub-problems and solving them in polynomial time.

III. Generation System

The generation process can logically be divided into two portions, ordering of the words in the target sentence and generating the morphological surface form of the words. A four step generation process is defined to address these two issues in the current work. Figure 2 illustrates these steps.

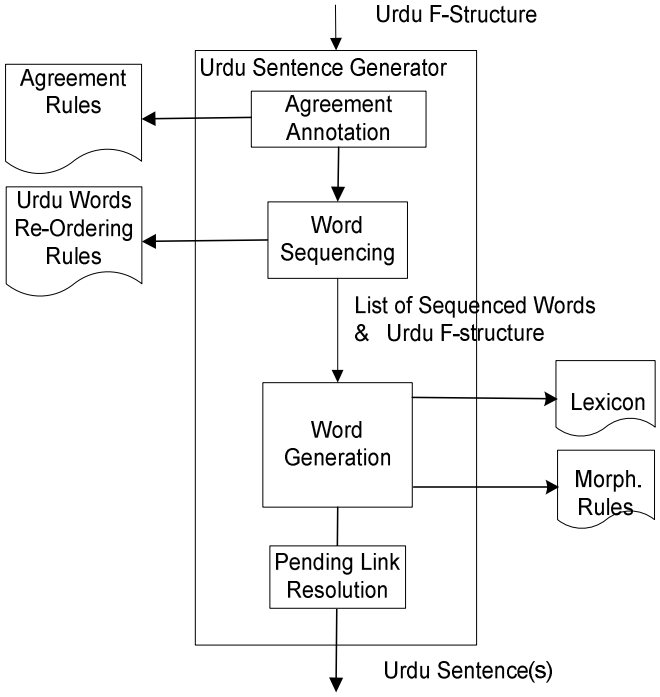

Figure 2: Revised Generator Architecture

The first, third and fourth steps collectively take care of the form of the words whereas the second step deals with the sequencing of the words, as summarized below and explained in the following sections.

Step 1: Some morphological features are decided at source language structure and are transferred to the Urdu f-structure through the Mapper, e.g. the number for nouns. Urdu has a very rich morphology as compared to English and in most cases words show more extensive agreements within phrases. Agreement annotation adds all such agreement based morphological features to the Urdu f-structure.

Step 2: Next comes the decision of the sequence of the words. Manually developed rules are used to define order of the words across f-structure nodes and order of words within these nodes according to Urdu language.

Step 3: Then a dictionary containing surface forms of the words and corresponding features is consulted to generate Urdu words.

Step 4: Finally, in case there are multiple words generated based on the f-structure features, a second pass on agreement rules is done in case any agreement rule is left unapplied in step one because features involving that rule were not present in the f-structure at that time. After dictio-

nary look up there is a possibility of addition of some new features which can trigger some more agreement rules.

Agreement Annotation

The Urdu f-structure received by the generator contains all the predicates and grammatical relations and a few features which can be derived from the English f-structure. For example, the Mapper gives an Urdu f-structure as shown in Figure 3 for the translation of the English sentence "He ate an apple."

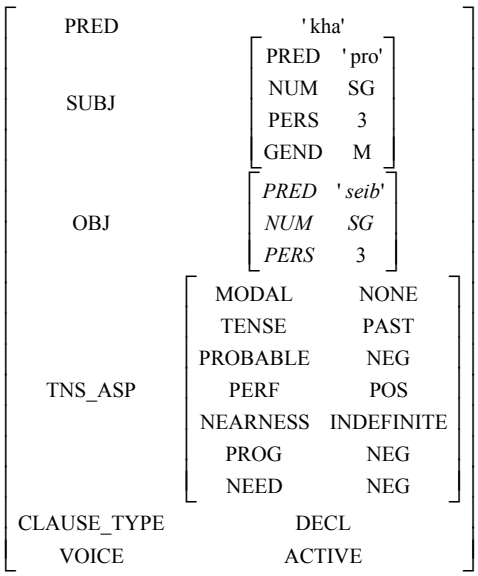

Figure 3: Urdu f-structure generated by the Mapper

More features are required to generate a correct Urdu sentence. For example, the verb 'kha' and its tense are mentioned in the f-structure but its number, gender and person are not specified. Furthermore, the case of the subject and object is also missing. In agreement annotation, these Urdu specific features are added in two steps.

First, all predicates in the f-structure are searched in Urdu lexicon and if there is any feature which has the same value for all the entries of a particular predicate, it is added to the f-structure. For example, the predicate 'seib' has three entries in the lexicon, singular 'seib', plural 'seib' and plural oblique 'seibon'. These three entries have the gender feature masculine. So GEND = M is added to the f-structure.

Second, more features are added according to the hand written agreement rules. These agreement rules examine features in the f-structure to add appropriate features.

The f-structure is traversed in a depth first manner and at each level the appropriate rule block is searched and applied to the f-structure if relevant. For example Table 1 states some rules for the ROOT block.

TABLE 1
SAMPLE AGREEMENT RULES

#	Rule	Explanation
1	^SUBJ CASE = !SUBJ_CASE IF (^TNS_ASP NEARNESS =c INDEFINITE).	Case of the subject should be same as governed by the verb when NEARNESS feature in TNS_ASP is INDEFINITE. It handles the past indefinite tense. SUBJ_CASE is feature present in all verbs which indicates the verb is ergative or nominative.
2	^SUBJ CASE = NOM.	Case of the subject should be nominative in all other cases.
3	^ _MORPH_FORM = PERFECTIVE IF (^TNS_ASP NEARNESS =c INDEFINITE).	Morphological form of the verb should be PERFECTIVE when NEARNESS is INDEFINITE.
4	^ _MORPH_FORM = HABITUAL IF ((^TNS_ASP TENSE =c PRES) && (^TNS_ASP PROG =c NEG)).	Morphological form of the verb should be HABITUAL for PRES tense.
5	!NUM = ^OBJ NUM, ! GEND = ^OBJ GEND, ! RESPECT = NORESPECT, !PERS = 3 IF ((((^TNS_ASP NEARNESS =c INDEFINITE) && (^ SUBJ_CASE =c ERG) && (^ OBJ NOUN =c POS) && (^ VOICE =c ACTIVE)))).	Verb should agree with OBJ for NUM, GEND, RESPECT and PERS features when NEARNESS = INDEFINITE and SUBJ_CASE = EGR.
6	!NUM = ^SUBJ NUM, ! GEND = ^SUBJ GEND, ! PERS = ^SUBJ PERS, ! RESPECT = ^SUBJ RESPECT.	Verb should agree with SUBJ for NUM, GEND, RESPECT and PERS features otherwise.

For the example of f-structure stated in Figure 3 the following rules will be applied.

- Rule # 1: Subject case should be ergative (ERG).
- Rule #3: Verb's morphological form (MORPH_FORM) should be perfective (PERF).
- Rule # 5: Verb should have number, gender, person and respect agreement with the object.

After applying the above rules, the output of the Agreement Annotation for the above mentioned f-structure is shown in Figure 4.

Word Sequencing

Sequencing rules take f-structure nodes and define their relative order. A hand written sequencing rule block is defined for each node of f-structure. Each rule block contains a sequencing order for the nodes that can possibly occur within that node.

The format of the rule block is as follows:

<NodeToBeSequenced>
[
<ChildNode1> : (<OptionalPOS>), <ChildNode2> : (<OptionalPOS>), … ? (<OptionalCondition>)
…
]

The rules also provide a list of possible parts of speech for the word at a particular node. The rule has an optional

condition part if there can be multiple orders for the same list of nodes.

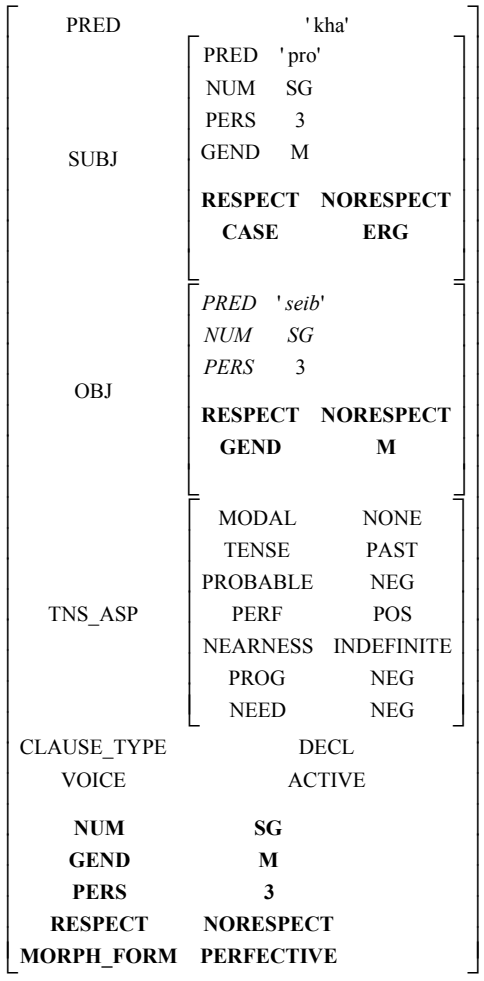

Figure 4: F-structure after Agreement

There are many nodes which require same ordering rules because of their linguistic similarity. For example, main ROOT node and COMP both are mainly based on verb phrase and their order will be same. SUBJ and OBJ both are noun phrase and follow the same sequence for their internal adjectives, determiners, specifiers, etc. To facilitate the rule writing process for such nodes, a macro rule concept is exploited. A macro rule is reused in all of the similar nodes.

These rules are applied on the target f-structure in depth-first manner. Sequencing rules for the f-structure mentioned in Figure 4 are as follows.

```
ROOT
[
        SUBJ: n,pro; (OBJ:n,pro;) PRED: v;
]

SUBJ
[
       PRED: n,pro;  (CASE:cm;)
]

OBJ
[
       PRED: n,pro;  (CASE:cm;)
]
```

The resultant of this process is a list of nodes and corresponding POS which indicate the desired sequence. The following table shows the output of above mentioned rules.

TABLE 2
OUTPUT OF WORD SEQUENCING MODULE

SUBJ. PRED	SUBJ. CASE	OBJ. PRED	ROOT. PRED
n,pro	cm	n,pro	V

Word Generation

The next step in the process is to generate the surface forms of the words from the sequenced list. Each element of sequenced list is traversed and lexicon is searched for the features present in that element along with the POS provided. If no match is found in the lexicon, value of the PRED feature in that element is assumed to be the surface word. If multiple words are found in the lexicon, they are listed along with the features found in the lexicon, to be eventually finalized in the next step. The corresponding result from Table 2 is given in Table 3.

TABLE 3
OUTPUT OF WORD GENERATION MODULE

SUBJ. PRED	SUBJ. CASE	OBJ. PRED	ROOT. PRED
n,pro	Cm	n,pro	V
اس	نے	سیب	کھایا

Pending Link Resolution

The sequenced word list is checked if there are any pending agreement rules which can now be evaluated on newly found features from the lexicon. After executing pending agreement rules, sentence(s) are generated by joining translation words.

The translation of the sentence will be as follows.

He ate an apple
اس نے سیب کھایا۔
[us ne seb khaya]

The following are a few sentences and their translations done by the system.

Who is serving as a spokesman for the student protesters
جو طالب علم احتجاجیوں لیے ترجمان کے طور پر خدمت کر رہا ہے
[jo talib-e-ilm ehtajajion liye tarjuman ke tor per khidmat kar raha hey]

Students must take part in security drills.
طالب علموں کو ضرور تحفظ مشقوں میں حصہ لینے چاہئیں

[talib-e-ilmon ko zaroor tahaffoz mashkon men hisa lene chahien]

The use of mobile phones by students in schools should be discouraged.
سکول میں طالب علموں کے پاس متحرک ٹیلیفون کے استعمال کی حوصلہ شکنی کی چاہیے
[skool men talib-e-ilmon ke pas mutaharik telefon ke istmal ki hoslashikni ki chahie]

IV. System Evaluation

The experiments reported in this paper are conducted on test data prepared from various news papers and websites. The sentences were selected randomly from news stories and articles in multiple domains including politics, religion, science, entertainment, weather etc. The test data contains 400 sentences, composed of 7410 words (with average sentence length of about 18 words). The random selection of test data covers a variety of grammatical structures.

Three reference translations have been prepared for the test data. Three translators are provided with the English sentences and they are requested to generate the Urdu translation independently. Some preprocessing is applied to all translations. The punctuations and diacritics are removed. This preprocessing is needed to ensure better comparison as different translators do not use diacritics and punctuations consistently.

The 400 test sentences are tested on both existing and new MT systems. Some of the test sentences could not be processed successfully; as the MT system hangs or runs out of memory. The system is evaluated using BLEU evaluation metric. The results of both systems are presented in Table 4.

TABLE 4
System Results

Old System			New System		
Successfully Processed Sentences	BLUE-1	BLUE-2	Successfully Processed Sentences	BLUE-1	BLUE-2
370	0.3	0.1	394	0.43	0.15

As shown in Table 4, the previous system processed 370 sentences successfully whereas for the remaining 30 sentences the system did not produce a result. In the new system only 6 sentences cannot be processed. The improvement is due to Collins parser [29] being used in place of the original parser. In addition, the translation accuracy is better in the new system as the BLUE scores depict.

The scores mentioned in Table 4 reflect the accuracy of machine translation systems. The BLUE scores computed for new system are 0.43 and 0.15 which reflect the performance of the new generation system in comparison with 0.3 and 0.1 of old system.

V. Discussion

The current system generates some output for any given f-structure even if there are no matching rules found. If the agreement rule is missing, the output will be generated without correct agreements. Further, the system generates output with default ordering of words. Since Urdu is a free phrase order language, the output without exactly correct order could be awkward but still somewhat understandable. In comparison, the previous system generated more accurate outputs for small sentences but in cases where there was no rule found for word order or agreement, the system was unable to generate any output. For example, following are the translations of a sentence by old and new systems.

There are two tragic anniversaries this month.
Old System: وہاں ہوتے ہوئے
[vahan hote hue]
New System: دو الم ناک سالگرہیں اس مہینے کو ہیں
[do alamnak salgirhen is maheene ko heyn]

The old system had no rule of this sentence structure so it just did the translation of a phrase in the sentence which does not convey the meaning of the sentence. Whereas the new system generated all the words and even though there is a grammatical error i.e. a wrong word کو, the meaning of the sentence is understandable.

Furthermore, the previous system tried to make a complete tree for the whole f-structure and if the grammar rule for a single phrase was missing, the whole sentence failed. The current system handles each level of f-structure separately which enables the system to generate partially correct outputs even when a small chunk of text is not correctly ordered.

Computational time and memory space required for generating complete trees and backtracking the rules is so much that sometimes system is unable to generate output for really long sentences i.e. with more than 40 words.

In some cases, one category in Urdu f-structure groups such phrases which require different sequencing in the sentence. For example, category ADJUNCT at verb level contains adverbs, adverbial phrases and negation adverb نہیں (NO). These different phrases may follow different sequence in Urdu sentence. For example, in following sentences bracketed phrases occur as ADJUNCT in f-structure. If the same sequencing rule is followed, the resultant Sentence 2 is not correct. Instead the different rule needs to be applied for correct output, as in Sentence 3.

1. اس نے [جلدی سے] کھانا کھایا۔
2. *اس نے [نہیں] کھانا کھایا۔
3. اس نے کھانا [نہیں] کھایا۔

Such classifications make the sequencing rules complicated or in some cases impossible to write if no distinguishing feature is present in the f-structure. Improvement in f-structure categories on the basis of error analysis can be helpful in improving the output of the system.

VI. Conclusion

In this paper we have presented a generation approach which subdivides the problem to improve the performance of the system. According to this methodology, morphologi-

cal features of words are decided according to agreement rules. Then the second issue of generation is addressed which is sequencing the words in correct order. This method has helped increase both the coverage and accuracy of translation, and lengthy sentences are also being generated successfully.

ACKNOWLEDGMENTS

This work has been carried through the initial support of Urdu Lexicon project of Ministry of IT, Govt. of Pakistan, at CRULP, NUCES and is now continuing through the support of PAN Localization project at CLE, KICS, UET (www.cle.org.pk).

REFERENCES

[1] S. Hussain, "Urdu Localization Project: Lexicon, MT and TTS", in the Proceedings of Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004, Geneva, Switzerland.

[2] R. M. Kaplan, K. Netter, J. Wedekind & A. Zaenen, "Translation by structural correspondences", in 'Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics, UMIST, Manchester, 10-12 April 1989, Association for Computational Linguistics, New Brunswick, NJ, pp. 272-281.

[3] N. Karamat, Verb Transfer for English to Urdu Machine Translation (Using Lexical Functional Grammar (LFG)). Unpublished MS Thesis, National University of Computer & Emerging Sciences, Lahore, Pakistan, 2006.

[4] R Quirk, et al.: Longman dictionary of contemporary English. Essex: Longman Dictionaries, 1987.

[5] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik, A Comprehensive Grammar of the English Language. Longman, 1985.

[6] I. Javed, Nayi Urdu Qawaid, Urdu Development Board, New Delhi, 1981.

[7] A. Haq, Qawaid-e-Urdu, Alnazir Press, Lakhnao, 1914.

[8] A. Haq, Urdu sarf va nahv, Anjuman-i Taraqqi Urdu Hind, Dehli, 1940.

[9] A. Siddiqui, Jamaul Qawaid (Comprehensive Grammar). Karachi, 1971.

[10] J. T. Platts, A Grammar of the Hindustani or Urdu Language, Sang-e-Meel Publications, 2002.

[11] T. Mohanan, Argument Structure in Hindi, Stanford University Center for the Study of Language, 1994.

[12] M. Butt, The structure of complex predicates in Urdu, Dissertations in linguistics, Center for the Study of Language (CSLI), 1995.

[13] H. Sarfraz and T. Naseem. Sentence Segmentation and Segment Reordering for English to Urdu Machine Translation. In the Proceedings of Conference on Language and Technology 2007. Bara Gali Summer Campus, University of Peshawar, Pakistan. 7-11 August 2007.

[14] U. Khalid, N. Karamat, , S. Iqbal, , S. Hussain, Semi-Automatic Lexical Functional Grammar Development, in the Proceedings of the Conference on Language and Technology 2009 (CLT09), FAST NU, Lahore, Pakistan, 22-24 Jan 2009 (URL: http://www.crulp.org/clt09/index.htm)

[15] U. Khalid, Semi-Automatic LFG Development System. Unpublished MS Thesis, National University of Computer & Emerging Sciences, Lahore, Pakistan, 2009.

[16] P. Whitelock, Shake and Bake Translation. In Proceedings of COLING 92, pages 610-616, Nantes, France, 1992.

[17] P. Whitelock, Shake-and-Bake Translation. In C. J. Rupp, M. A. Rosner, and R. L. Johnson, editors, Constraints, Language and Computation, pages 339-359. Academic Press, London, 1994.

[18] J. L. Beaven, Lexicalist Unification-based Machine Translation. Ph.D. Thesis, University of Edinburgh, Edinburgh, 1992.

[19] J. L. Beaven, Shake-and-Bake Machine Translation. In Proceedings of COLING 92, pages 602-609, Nantes, France, 1992.

[20] F. Popowich, Improving the Efficiency of a Generation Algorithm for Shake and Bake Machine Translation using Head-Driven Phrase Structure Grammar. Technical Report CMPTTR 94-07, School of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada, 1994.

[21] C. Brew, Letting the Cat out of the Bag: Generation for Shake-and-Bake MT. In Proceedings of COLING 92, pages 29-34, Nantes, France, 1992.

[22] P. F. Brown , J. Cocke , S. A. Della Pietra , V. J. Della Pietra , F. Jelinek , J. D. Lafferty , R. L. Mercer , P. S. Roossin, A statistical approach to machine translation, Computational Linguistics, v.16 n.2, p.79-85, June 1990

[23] H. Chen and Y. Lee, A Corrective Training Algorithm for Adaptive Learning in Bag Generation. In International Conference on New Methods in Language Processing (NeMLaP), pages 248-254, Manchester, UK. UMIST, 1994.

[24] V. Poznański , J. L. Beaven , P. Whitelock, An efficient generation algorithm for lexicalist MT, Proceedings of the 33rd annual meeting on Association for Computational Linguistics, p.261-267, June 26-30, 1995, Cambridge, Massachusetts

[25] W. Wang , K. Knight , D. Marcu, Capitalizing machine translation, Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, p.1-8, June 04-09, 2006, New York, New York

[26] E. Minkov, K. Toutanova, Hisa, Generating complex morphology for machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07) 2007.

[27] K. Toutanova and H. Suzuki. Generating case markers in machine translation. Association for Computational Linguistics, 2007.

[28] K. Toutanova, H. Suzuki, and A. Ruopp. Applying morphology generation models to machine translation Proceedings of ACL-08, 2008.

[29] M. Collins, Head-Driven Statistical Models for Natural Language Parsing. PhD Dissertation, University of Pennsylvania, 1999.