A Pattern Matching Approach for Redundancy Detection in Bi-lingual and Mono-lingual Corpora

Muneer Ahmad, Hassan Mathkour

Abstract----The Bi-Lingual and Mono-Lingual Corpora Information relating to numerous Languages may be duplicated. This leads to slow and inaccurate search results from Bi-Lingual and Mono-Lingual databases. It is essential to structure the Sequences in a fashion that reduces the redundant sequence structure so that the analysis of Bi-Lingual and Mono-Lingual Corpora structure is accurate to help in analyzing the features of certain complex and subjective languages. The detection will lead to the selection of right solution from large Corpora's.

In this paper, we present an algorithm (we call it DSDR) that operates on a set of Bi-Lingual and Mono-Lingual Corpora and iterates in the same set to find all possible duplications present in the set. Once the duplications are found, the DSDR removes duplicated Chains and refreshes the databases resulting in remarkable reductions in the sizes of the databases. In addition, the speed of searches of certain Chains from Bi-Lingual and Mono-Lingual Corpora becomes quite fast and accurate.

Key Words: **Bi-Chains, Corpora, DSDR, Mono-Chains, Sequences**

I. Introduction

Lexical resources are necessary for any type of natural language processing and language engineering applications. Where in the early days of language engineering lexical information may have been hard-coded into the system, today most systems and applications rely on explicitly introduced and modularly designed lexica to function: examples range from applications such as automatic speech recognizers, dialogue systems, information retrieval, and writing aids to computational linguistic techniques such as part-of-speech tagging, automatic thesaurus construction, and word sense disambiguation systems [16].

Manuscript received November 12, 2008. This work was supported by the Prince Sultan Bin Abdul Aziz Research Program and Research Center College of Computer and Information Sciences King Saud University.

Muneer Ahmad is affiliated with Department of Computer Science, College of Computer and Information Sciences, King Saud University P.O. Box 51178, Riyadh 11543, Saudi Arabia (muneerahmadmalik@yahoo.com)

Hassan Mathkour is affiliated with Department of Computer Science, College of Computer and Information Sciences, King Saud University P.O. Box 51178, Riyadh 11543, Saudi Arabia (binmathkour@yahoo.com)

The Corpora aids almost all the natural language processing tools from token analysis to speech recognition systems, definitely these tools are backed by Corpus. The modern machine learning systems like Trado's and EBMT (example based machine translation) comprehensively use translated phrases and un-translated are stored as chunks of tokens (that are latterly translated). Broadly speaking, lack of refinement leads to garbage collection and garbage reduces performance of systems. The proposed approach in this paper is a strong motivation for making these tools fast and accurate.

The first step is to find the best alignment in sequences namely local and global, In global alignment, we need to find the best alignment for the entire sequences or the set of sequences, for local alignment this attempt is confined to certain small regions / characters on which the alignment is desired, while in multi-alignment, more than two sequences are globally aligned. We may use a scoring scheme for evaluation of matching letters, no of mis-matching letters, the goal of making such scores is to produce a resultant optimal solution that best reflect the alignment problem. The Multiple Sequence Alignment is a sequence alignment of three or more corpora chains. This kind of alignment helps to better understand the relevancy between corpora chains.

Multiple Sequence Alignment is used for many reasons, namely,

1. Discovering the regions where similarity and differences can be found
2. To provide the degree of strength by introducing gaps in specific regions
3. The MSA algorithms help to better provide the chances of predicting matches and mis-matches that brings more related results between species.

Nowadays, multiple sequence alignment is an important tool that provides key information for sequence analysis. There are several uses of MSA; finding sequence to determine patterns that characterize sequence patterns; detecting homology between new sequences and known already existing chains in corpora structure.

The objective of this approach is to help the computer systems that use computational linguistics techniques. These linguistic systems mostly rely on relative Corpus such as automatics thesaurus construction and word sense problems. The great concentration is taken into account to provide robustness for the efficiency and reliability of such systems. Some times it may happen that large corpus chains themselves contain sub-chains that serve as a burden to corpus and provide no useful information, at

the first instance, the data bank is traversed by such large chains and then the large chains are fragmented for further refinements.

The prior techniques are either malfunctioning or slow. The need of time is to provide better solutions to underlying problem.

II. Previous Work

The following techniques are being used for the alignment of two sequences [1, 2].

A. DOT MATRIX method.
B. The Dynamic Programming method.
C. WORD or k-tupple methods.

A. DOT MATRIX Method

The DOT MATRIX method is useful only when sequences are known to be very much alike. This is because it displays any possible sequence alignment as diagonals on the matrix. It may be used for insertion / deletion and direct / inverted repeats of characters of the sequences. The Major limitation of this method is that most DOT MATRIX programs do not show an actual alignment. [2]. Figure 1 depicts an example.

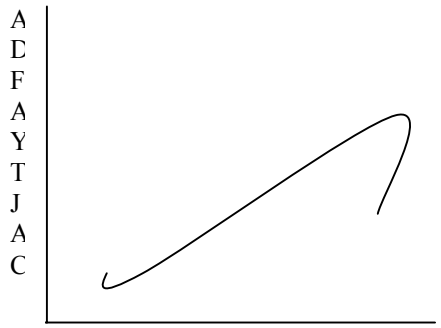

Fig.1 DOT MATRIX APPROACH

B. Dynamic Programming Method

The Dynamic Programming Method is mostly used for Global Alignment of sequences. It was devised by Needleman and Wunsch in 1970. It has been used for Local Alignment by Smith and Waterman in 1981. The procedure of this method attempts to match all possible pairs of characters [5] between sequences and adopts a scoring scheme for matches, mismatches and gaps. This method is widely used for both kinds of alignments. However, it has a major drawback that it can be slow due to very large number of computational steps which increase approximately as square cube of sequence lengths. Thus utilization of this method for large sequences is not feasible [1]. The Dynamic Programming Method used for Global Alignment of a pair of sequences can be extended for Multiple Sequence Alignment. But the limitation of this method is that it can not efficiently align more sequences, when the no. of sequences grows, the performance of the method degrades considerably.

Progressive Methods [5] use the Dynamic Programming Method to built the MSA (Multiple Sequence Alignment) starting with most related sequences and then progressively adding less related sequences to initial alignment.

Examples [5]

CLUSTALW b) PILEUP

The drawbacks of Progressive Methods are dependent of initial pair-wise Sequence Alignment. The very first sequences must be very closely related sequences, if sequences are closely aligned then there will be few errors but if sequences are not closely aligned there will be more errors.

Iterative Methods [6] attempt to correct for the problem raised by Progressive Methods by repeatedly realigning subgroups of sequences and then by aligning these subgroups into Global Alignment [6, 7]

C. WORD or K-Tupple Methods

The WORD or K-Tupple Methods are used by the FASTA and BLAST algorithms [1, 2]. They align two sequences by first searching for identical parts of sequences and then joining them for alignment purpose by Dynamic Programming Methods. These methods can be reliable in computational and statistical sense bringing accurate results, but they are slow.

III. The proposed Algorithm: DSDR

The Variables

i, x, z, k → Loop Variables
status → checks status of current Chain
location→ specifies Location in Corpora
flag→ Decision variable
count→ Bi-Lingual and Mono-Lingual Corpora counter

Suppose

G = {g1, g2, g3, g4, --------- gN}

Where G is a set of all possible Bi-Lingual and Mono-Lingual Corpora Chains Sequences g1, g2, g3, g4, ----- gN

Repeat for g1 to gN
Repeat for i = 1 to I = N
IF gi equals Tj ε G then
    Set location = i
    Var status = 1
    Var z = i
Repeat for x = 1 to length (gk)

```
                K = 2, 3, 4, ---, N
                Z = z+1
        IF gk equals Tm ε G then
                Set status = 1
        ELSE
                Set status = 0

                END INNER REPEAT
                    END IF
                      END IF
                END IF

        IF status equals 1 then
                Set count = count +1
                    flag = 1
    IF Occurrence greater than 0 then
    Repeat for k = I to k < I + length (gm)
    gm ε G

    G [g] k = G [g] k + length (gm)

                END REPEAT
                END IF

        Occurrence = Occurrence + 1
        Status = 0
                END IF
                END OUTER REPEAT

    -- END DSDR PROCEDURE
```

A. Complexity of DSDR

The complexity of DSDR is N(N-1)*log (n)
Where
N→ No. of Bi-Lingual and Mono-Lingual Corpora Chains
n→No. of Bi-Lingual and Mono-Lingual Corpora Sub-Chains

The Big O notation of DSDR is dependant upon the possible input size N of Corpus and sub-chain's strength n, the complexity grows with the input size. The iterations for the Corpus and its subsequent chains lead to N (N-1) with addition of refinements log (n) for sub-chains.

B. Functionality

The DSDR takes two input parameters:

Size of G
Total no. of Bi-Lingual and Mono-Lingual Corpora Chains

Size of Chains
-No. of Characters in one Chains

The algorithm is initially run for N Corpora Chains in the form of main outer loop that iterates and moves among the Chains of Characters from Corpora present in the set G of total inputs. There may be several duplicated Chains in the same set G, the algorithm finds and removes them from the set providing the updated copy of G.

IV. Corpora Design Structure

The Corpora may serve as database or the data warehouse depending upon the criteria set for fragmenting input data,Source data is read into a file and fragmented into clauses at specific cutting points, e.g Comma (,), Colon (:), Full Stop (.), Semi Colon (;) etc. Normally the clauses are vectored to get the appropriate meanings from memory; the un-found clauses are kept separate for later use. Words are the smallest sequence in this order, the un-found words may be analyzed to provide more refined results. Clause boundaries must be mentioned while designing a sophisticated warehouse, possible clause boundaries may be,

"a", "above", "after", "am", "an", "and", "any", "are", "as", "at", "because", "beyond", "by", "did", "do", "does", "except", "for", "from", "has", "if", "in", "is", "may", "nor", "nor", "of", "on", "or", "over", "shall", "since", "so", "such", "than", "that", "them", "then", "those", "till", "to", "under", "until", "up to", "was", "were", "what", "when". "where", "where as", "who", "whom", "whose", "will", "with", "within"

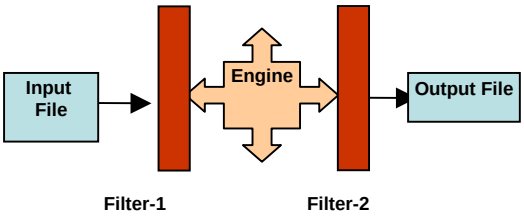

Fig. 2 Primary Filter

The warehouse of this corpora receives the input data, passes it through first filter, this filter generates token sequence, generate fragment from tokens and analyze the fragments,

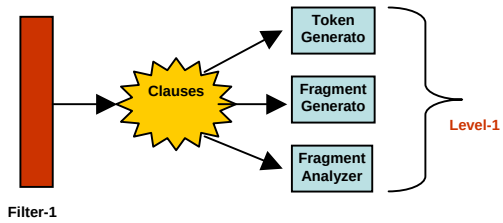

Fig.2 Secondary Filter

The warehouse corpora is considered to be level-1 repository, this repository is not necessarily permanent storage of sequences, rather holds the structures only for data manipulation till certain time when preprocessing is considered to be complete.

The preprocessing stage always works as data cleansing for better optimal sequence manipulation, now consider level-2

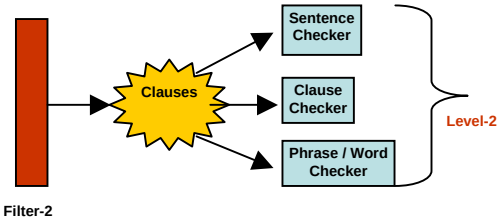

Fig. 3 Tertiary Filter

The fragmented sequences are analyzed, possible clauses are checked and even low level phrase / word checking is also made, the intermediate engine serves as a coordination channel between the two filters, the target file is generated after the data is scanned and analyzed from the second filter.

The transitivity condition is only applicable when some parts or fragments of sequences need vector (directing to some other language) attributes.

V. Example Based Repositories

A. Monolingual Corpora (Mono)

The Monolingual Corpora is self contained i.e the sequences in corpora don't necessarily reflect the corresponding some other directed sequences, so it is relatively easy to overcome redundancy in this corpus. For instance, consider the example,

"This report is an attempt to make workable recommendations to the Govt. of KSA for bringing some advance suitable reforms for the progress of culture and civilization"

If the above sentence is to pass through a language chopper, following will be the possible sequence clauses

"This report", "is an attempt", "to make workable recommendations", "to the Govt. of KSA", "for bringing some advance suitable reforms", "for the progress", "of culture and civilization"

Imagine that the chopper receives a similar sentence that may contain some repeated sequence structure, the structure will be stored in the database, if the size of input data is becoming huge then the size of corpora also increases, the need is felt to frame a suitable method that performs regular checks representing the cleaning of corpora.

B. Bi-Lingual Corpora

In case of bi-lingual corpora, the sequence data is not self-contained and need vector attributes; this is considered to be the most difficult phenomenon as memory needs address based references, the corresponding parts of the sequences are read from this repository, this direction is similar as in case of mono-lingual corpora, once the chains are found, the redundancy is removed in the same fashion. So we can conclude that bi-lingual warehouse is one step far from the mono one.

For bringing more optimal results, Computational Grammar and Morphological rules are required; so far there have been no defined rules for exact placement of clauses, adjustment of induction and grammatical issues.

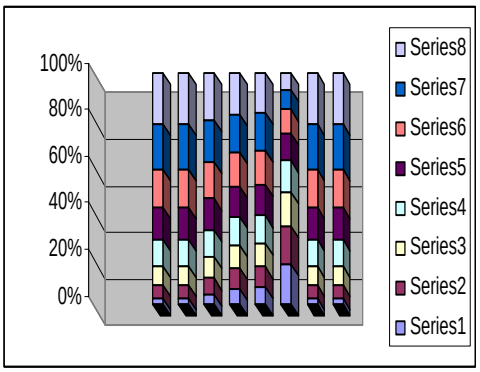

Fig. 4 Graphical Representation of Results

The Graph show that as the sequences grow in size then approximate matching may also be positive and vise versa. Series 6, 7 and 8 are obvious from their behavior that with the extend of Chains from Corpora, the probability of matching definitely increases, but it can not be said that as we make the sequences too lengthy, the results would be according to expectations, for instance analyze the following sequences with given data

Sequence→ A

ASCVFTHUJKOCAWCT….ATCQETGFY..UATC..AA
TCSWRTYUTTCAPTCAGCT….ATCGAREQA..GAT
C..AATCXCVBNM,TCAGTCAGCT….ATCG……..AT
GCC..GATC..AATCGGCATGTTCAGTCAGCT….ATC
GATGCC..GATC..AATCGQWERTYUIOPAPTCAGCT
….ATCGAREQA..GATC..AATCXCVBNM,TCAGTCA
GCT….ATCG……..ATGCC..GATC..AATCGGCATGT
TCAGTCAGCT….ATCGATGCC..GATC..AATCGQW
ERTYUIOPCVBNM,TCAGTCAGCT….ATCG……..A
TGCC..GATC..AATCGGCATGTTCAGTCAGCT….AT
CGATGCC..GATC..AATCGQWERTYUIOPAPTCAGC
T….ATCGAREQA..GATC..AATCXCVBNM,TCAGTC
AGCT….ATCT….ATCGATGCC..GATC..AATCGQW
ERTYUIOPCQA..GATC..AATCXCVBNM,TCAGTCA
GCT….ATCG……..ATGCC..GATC..AATCGGCATGT
TCAGTCAGCT….ATCGATGCC..GATC..AATCGQW
ERTYUIOPCVBNM,TCAGTCAGCT….ATCG……..A
TGCC..GATC..AATCGGCATGTTCAGTCAGCT….AT
CGATGCC..GATC..AATCGQWERTYUIOPAPTCAGC
T….ATCGAREQA..GATC..AATCXCVBNM,TCAGTC
AGCT….ATCT….ATCGATGCC..GATC..AATCGQW
ERTYUIOPCVBNM,TCAGTCAGCT….ATCG……..A
TGCC..GATC..AATCGGCATGTTCAGTCAGCT….AT
CGATGCC..GVBNM,TCAGTCAGCT….ATCG……..A
TGCC..GATC..AATCGGCATGTTCAGTCAGCT….AT
CGATGCC..GATC..AATCGQWERTYUIOPAPTCAGC
T….ATCGACG……..ATGCC..GATC..AATCGGCATG
TTCAGTCATCAGCT….ATCG…….

Sequence→ B

AAWERT..YATC..AATCGGCATGTTCAGTCAGCT…
.ATCFATYYC..IATO..AATCQIOATGAWERTTCAGC
T….ATCGATUYTREQATC..AATCGGCAQWERTFD
SAAGCT….ATCG……..ATCVBN.GATC..AATCMNB
VCGFDCAGATC..AATCGGCATGTTCAGTCAGCT…
.ATCFATYYC..IATO..AATCQIOATGAWERTTCAGC
T….ATCGATUYTREQATC..AATCGGCAQWERTFD
SAAGCT….TCAGCT….ATCC..GATC..AATCSERTH
YJUTCAGTCAGCT….AGGCATGTTCAGTCAGCT….
ATCFATYYC..IATO..AATCQA..GATC..AATCXCVB
NM,TCAGTCAGCT….ATCG……..ATGCC..GATC..A
ATCGGCATGTTCAGTCAGCT….ATCGATGCC..GA
TC..AATCGQWERTYUIOPCVBNM,TCAGTCAGCT
….ATCG……..ATGCC..GATC..AATCGGCATGTTCA
GTCAGCT….ATCGATGCC..GATC..AATCGQWERT
YUIOPAPTCAGCT….ATCGAREQA..GATC..AATCX
CVBNM,TCAGTCAGCT….ATCT….ATCGATGCC..G
ATC..AATCGQWERTYUIOPCVBNM,TCAGTCAGCT
….ATCG……..ATGCC..GATC..AATCGGCATGTTCA
GTCAGCT….ATCGATGCC..GQIOATGAWERTTCA
GCT….ATCGATUYTREQATC..AATCGGCAQWERT
FDSAAGCT….TCAGCT….ATCC..GATC..AATCSER
THYJUTCAGTCAGCT….AATCQIOATGAWERTTCA
GCT….ATCGATUYTREQATC..AATCGGCAQWERT
FDSAAGCT….TCAGCT….ATCC..GATC..AATCSER
THYJUTCAGTCAGCT….AGGCATGTTCAGTCAGC
T….ATCFATYYC..IATO..AATCQIOATGAWERTTC
AGCT….ATCGATUYTREQATC..AATCGGCAQWER
TFDSAAGCT….TCAGCT…TCG……

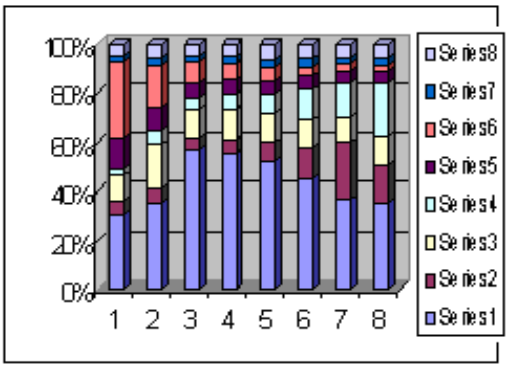

Fig. 5 Graphical Results for Larger Sequences

The phenomenon is quite obvious from the graphical results that as we increase the sequence size enormously then matching tendency also decreases, so it is mandatory to keep the sequences at some standard specified lengths mentioned in Corpora for optimum results.

VI. Conclusion

DSDR is an algorithm for finding and removing Duplicate Sequence of Bi-Lingual and Mono-Lingual Corpora Chains in Bi-Lingual and Mono-Lingual Corpora. This would greatly reduce the overhead involved in time consuming and slow searches of certain Chains in Large Corpora.

The algorithm operates on a set of Bi-Lingual and Mono-Lingual Corpora Chains and iterates in the same set to find all possible duplications present in the set, once the duplications are found, the DSDR removes duplicated Chains and refreshes the databases resulting in remarkable reduction in size of databases and also the speed of searches of certain Chains from Bi-Lingual and Mono-Lingual Corpora becomes quite fast and accurate.

VII. Acknowledgements

This work was supervised by Director Research Center, College of Computer and Information Sciences, King Saud University Riyadh Saudi Arabia under research program entitled "Prince Sultan Bin Abdul Aziz research Program for Distinguished researchers".

VIII. References:

[1]. Gale, W. and K. Church, Identifying word correspondences in parallel texts in Proceedings of the DARPA Workshop on Speech and Natural Language, Pacific Grove, California ,Pages: 152 –157, 1991.

[2]. Bingham, Ella and Heikki Mannila, Random projection in dimensionality reduction in International Conference on Knowledge Discovery and Data Mining, Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining San Francisco, California, Pages: 245 – 250, 2001, ISBN:1-58113-391-X

[3]. Achlioptas, Dimitris, Database friendly random projections, Symposium on Principles of Database Systems, Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Santa Barbara, California, United States, Pages: 274 – 281, 2001, ISBN:1-58113-361-8

[4]. Brown, Peter F., John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, A statistical approach to machine translation in Computational Linguistics, 16(2):79–85, 1990

[5]. Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman, Indexing by latent semantic analysis in Journal of the Society for Information Science, 41(6):391–407, 1990

[6]. Gale, W. and K. Church, Identifying word correspondences in parallel texts in Proceedings of the DARPA Workshop on Speech and Natural Language Pacific Grove, California 1991

[7]. Grefenstette, G., Evaluation techniques for automatic semantic extraction: Comparing syntactic and window-based approaches in Corpus processing for lexical acquisition**,** Pages: 205 – 216, 1996, ISBN:0-262-02392-X, 1993

[8]. Hecht-Nielsen, R., Context vectors: general purpose approximate meaning representations self-organized from raw data. In J.M. Zurada, R.J. Marks II, and C.J. Robinson (eds.), Computational Intelligence: Imitating Life. IEEE Press, pages 43–56, 1994

[9]. Johnson, W.B. and J. Lindenstrauss, Extensions of lipshitz mapping into Hilbert space in Contemporary Mathematics, 26:189–206, 1984

[10]. Kanerva, P., J. Kristofersson, and A. Holst, 2000. Random indexing of text samples for latent semantic analysis in

Proceedings of the 22nd Annual Conference of the Cognitive Science Society. Erlbaum, pages 103-106, 2000

[11]. Karlgren, H., J. Karlgren, M. Nordstrom, P. Pettersson, and B. Wahrol´en, Dilemma – an instant lexicographer in Proceedings of the 15th Annual Conference on Computational Linguistics (COLING 94) 1994

[12]. Kaski, S., Dimensionality reduction by random mapping: Fast similarity computation for clustering in Proceedings of the IJCNN'98, International Joint Conference on Neural Networks. IEEE Service Center, 1998

[13]. Landauer, T. and S. Dumais, A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychological Review, 104(2):211–240, 1997

[14]. Sahlgren, M., Automatic bilingual lexicon acquisition using random indexing of aligned bilingual data in Proceedings of the fourth international conference on Language Resources and Evaluation, LREC 2004

[15]. M. SAHLGREN and J. KARLGREN, Automatic Bilingual Lexicon Acquisition Using Random Indexing of Parallel Corpora Swedish Institute of Computer Science, SICS Box 1263, SE-164 29 Kista, Sweden {mange, jussi}@sics.se