The South African Institute of Computer Science and Information Technology

The 1997 National Research and Development Conference

Riverside Sun
Vanderbijlpark
13 & 14 November

Hosted by
Potchefstroomse Universiteit
vir Christelike Hoër Onderwys

The Department of Computer Science and Information Systems
Potchefstroom University for Christian Higher Education
Vaal Triangle Campus

PROCEEDINGS

Edited by L.M. Venter & R.R. Lombard
The South African Institute of Computer Science
and
Information Technology

Proceedings
of the
The 1997 National
Research and
Development
Conference
Towards 2000

Riverside Sun
Vanderbijlpark
13 & 14 November

Edited by
L.M. Venter
R.R. Lombard
Foreword

This book contains a collection of papers presented at a Research and Development conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT). The conference was held on 13 & 14 November 1997 at the Riverside Sun, Vanderbijlpark. Most of the organization for the conference was done by the Department of Computer Science and Information Technology of the Vaal Triangle Campus, Potchefstroom University for Christian Higher Education.

The programming committee accepted a wide selection of papers for the conference. The papers range from detailed technical research work to reports of work in progress. The papers originate mainly from Academia, but also describe work done in and for Industry. It is hoped that the papers give a true reflection of the current research scene in Computer Science and Information Technology in South Africa. Since one of the aims of the conference is Research development, the papers were not subjected to a refereeing process.

A number of people spent numerous hours helping with the organization of this conference. In this regard, we wish to thank the members of the Organizing committee, and the Programming committee who had very little time to screen the abstracts and compile the program. A special thanks goes to the secretary of the department, Mrs Helei Jooste, whose very able work was interrupted by the birth of her first child.
Organizing Committee

Conference General Chairs
Prof. J.M. Hattingh (PU for CHE)

Organizing Chair
Prof. Lucas Venter (PU for CHE)

Organizing Committee
Mrs. S. Gilliland
Mr. J.P. Jooste
Mr. R.R. Lombard
Mrs. M. Huisman

Secretariat
Mrs. H. Jooste

Program Chair
Prof A de Waal (PU for CHE)

Program Committee
Prof D. Kourie (UP)
Prof. C. Bornman (UNISA)
Prof. L.M. Venter (PU for CHE)
### Table of Contents

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreword</td>
<td>i</td>
</tr>
<tr>
<td>Organizing Committee</td>
<td>ii</td>
</tr>
<tr>
<td>List of Contributors</td>
<td>vii</td>
</tr>
<tr>
<td><em>Software Objects Change: Problems and Solution</em></td>
<td>1</td>
</tr>
<tr>
<td>S.A. Ajila</td>
<td></td>
</tr>
<tr>
<td><em>Liming-like Curve Constructions</em></td>
<td>26</td>
</tr>
<tr>
<td>M.L. Baart and R. McLeod</td>
<td></td>
</tr>
<tr>
<td><em>A Model for Evaluating Information Security</em></td>
<td>27</td>
</tr>
<tr>
<td>L. Barnard and R. von Solms</td>
<td></td>
</tr>
<tr>
<td><em>Integrating Spatial Data Management and Object Store Technology</em></td>
<td>31</td>
</tr>
<tr>
<td>S. Berman, S. Buffler and E. Voges</td>
<td></td>
</tr>
<tr>
<td><em>Metamodelling in Automated Software Engineering</em></td>
<td>32</td>
</tr>
<tr>
<td>S. Berman and R. Figueira</td>
<td></td>
</tr>
<tr>
<td><em>Using Multimedia Technology for Social Upliftment in Deprived Communities of Southern Africa</em></td>
<td>33</td>
</tr>
<tr>
<td>L. Bester and E. de Preez</td>
<td></td>
</tr>
<tr>
<td><em>Extending the Client-Server Model for Web-based Execution of Applications</em></td>
<td>36</td>
</tr>
<tr>
<td>L. Botha, J.M. Bishop and N.B. Serbedzija</td>
<td></td>
</tr>
<tr>
<td><em>Access Control Needs in an Electronic Workflow Environment</em></td>
<td>45</td>
</tr>
<tr>
<td>R.A. Botha</td>
<td></td>
</tr>
<tr>
<td><em>The Use of the Internet in an Academic Environment to Commercially Supply and Support Software Products</em></td>
<td>51</td>
</tr>
<tr>
<td>B. Braude and A.J. Walker</td>
<td></td>
</tr>
<tr>
<td><em>Explanation Facilities in Expert Systems Using Hypertext Technology</em></td>
<td>63</td>
</tr>
<tr>
<td>T. Breetzke and T. Thomas</td>
<td></td>
</tr>
<tr>
<td><em>Theoretical Computer Science: What is it all about, and is it of any relevance to us?</em></td>
<td>75</td>
</tr>
<tr>
<td>C. Brink</td>
<td></td>
</tr>
<tr>
<td><em>Representing Quadrics on a Computer</em></td>
<td>76</td>
</tr>
<tr>
<td>M.A. Coetzee and M.L. Baart</td>
<td></td>
</tr>
</tbody>
</table>
The Generation of Pre-Interpretations for Detecting Unsolvable Planning Problems
D.A. de Waal, M. Denecker, M. Bruynooghe and M. Thielscher

The Emerging Role of the Chief Information Officer in South Africa
B. Dekenah

A Java-Implemented Remote Respiratory Disease Diagnosis System on a High Bandwidth Network
A. Foster

Early Results of a Comparative Evaluation of ISO 9001 and ISO/IEC 15504 Assessment Methods Applied to a Software Project
C. Gee and A.J. Walker

A Neural Network Model of a Fluidised Bed
M. Hajek

The Effects of Virtual Banking on the South African Banking Industry
M.L. Hart and M. Dunley-Owen

Linear Response Surface Analysis and Some Applications
J.M. Hattingh

Model Checking Software with Symbolic Trajectory Evaluation
A. Hazelhurst

A Risk Model to Allocate Resources to Different Computerized Systems
H.A. Kruger and J.M. Hattingh

Returns on the Stock Exchange
J.W. Kruger

Cardinality Constrained 0-1 Knapsack Problems
M.F. Kruger, J.M. Hattingh and T. Steyn

An Investigation in Software Process Improvement in the Software Development of a large Electricity Utility
M. Lang and A.J. Walker

Design and Implementation of a C++ Package for Two-Dimensional Numerical Integration
D.P. Laurie, L Pluym and Ronald Cools

Algebraic Factorization of Integers Using BDE's
H. Messerschmidt and J. Robertson
Global Optimization of Routes after the Process of Recovery
M. Mphahlele and J. Roos

Using a Lattice to Enhance Adaptation Guided Retrieval in Example Based Machine Translation
G.D. Oosthuizen and S.L. Serutla

Information Systems Development and Multi Criteria Decision Making / Systems Thinking
D. Petkov, O. Petkova

The Development of a Tutoring System to Assist Students to Develop Answering Techniques
N Pillay

Combining Rule-Based Artificial Intelligence with Geographic Information Systems to Plan the Physical Layer of Wireless Networks in Greenfield Areas
K. Prag, P. Premjeeth and K. Sandrasegaran

A Distributed Approach to the Scheduling Problem
V. Ram and P. Warren

More readings than I thought: Quantifier Interaction in Analysing the Temporal Structure of Repeated Eventualities
S. Rock

Ray Guarding Configuration of Adjacent Rectangles
I. Sanders, D. Lubinsky and M. Sears

Developing Soft Skills in Computer Students
C. Schröder, T. Thomas

Information Security Awareness, a Must for Every Organization
M. Thomson and R. von Solms

Pla Va: A Lightweight Persistent Java Virtual Machine
S. Tjasink and S. Berman

Beliefs on Resource-Bounded Agent
E. Viljoen

Object-Orientated Business Modelling and Re-engineering
M. Watzenboeck
On Indexing in Case Based Reasoning Applied to Pre-Transportation Decision Making for Hazardous Waste Handling
K.L. Wortmann, D. Petkov and E Senior

Author Index
# List of Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution and Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.A. Ajila</td>
<td>Department of Mathematics and Computer Science, National University of Lesotho, Roma, 180, Lesotho</td>
</tr>
<tr>
<td>L. Baart</td>
<td>Department of Mathematics, Vaal Triangle Campus of the PU for CHE, PO Box 1174, Vanderbijlpark, 1900</td>
</tr>
<tr>
<td>L. Barnard</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>S. Berman</td>
<td>University of Cape Town, Rondebosch, 7701</td>
</tr>
<tr>
<td>L. Bester</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>J.M. Bishop</td>
<td>Computer Science Department, University of Pretoria, Pretoria, 0002</td>
</tr>
<tr>
<td>L. Botha</td>
<td>Computer Science Department, University of Pretoria, Pretoria, 0002</td>
</tr>
<tr>
<td>R.A. Botha</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>N. Braude</td>
<td>Software Engineering Applications Laboratory, Electrical Engineering, University of the Witwatersrand, Private Bag 3, Wits, 2050</td>
</tr>
<tr>
<td>T. Breetzke</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>C. Brink</td>
<td>University of Cape Town, Rondebosch, 7700</td>
</tr>
<tr>
<td>M. Bruynooghe</td>
<td>Departement Computerwetenschappen, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium</td>
</tr>
<tr>
<td>S. Buffler</td>
<td>University of Capetown, Rondebosch, 7701</td>
</tr>
<tr>
<td>M.A. Coetzee</td>
<td>Department of Mathematics, PU for CHE, Private Bag X6001, Potchefstroom, 2520</td>
</tr>
<tr>
<td>R. Cools</td>
<td>Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium</td>
</tr>
<tr>
<td>E. de Preez</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>L. Bester</td>
<td>Faculty of Computer Studies, Port Elizabeth Technikon, Private Bag X6011, Port Elizabeth, 6000</td>
</tr>
<tr>
<td>M. Denecker</td>
<td>Departement Computerwetenschappen, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium</td>
</tr>
<tr>
<td>M. Dunley-Owen</td>
<td>Department of Information Systems, University of Cape Town, Rondebosch, 7700</td>
</tr>
<tr>
<td>R. Fiqueira</td>
<td>University of Cape Town, Rondebosch, 7701</td>
</tr>
<tr>
<td>A. Foster</td>
<td>Department of Computer Science, University of Cape Town, Rondebosch, 7701</td>
</tr>
<tr>
<td>C. Gee</td>
<td>Software Engineering Applications Laboratory, Electrical Engineering, University of the Witwatersrand, Private Bag 3, Wits 2050</td>
</tr>
</tbody>
</table>

vii
M. Hajek
Department of Computer Science
University of Durban Westville
Private Bag X54001
Durban, 4000

M.L. Hart
Department of Information Systems
University of Cape Town
Rondebosch, 7700

J.M. Hattingh
Department of Computer Science and
Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2520

S. Hazlehurst
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits, 2050

H.A. Kruger
Department of Computer Science and
Information Systems
PU for CHE
Private Bag X6001
Potchefstroom, 2520

J.W. Kruger
University of the Witwatersrand
Private Bag 3
Wits, 2050

M.F. Kruger
PU for CHE
Private Bag X6001
Potchefstroom, 2520

M.T. Lang
Eskom Information Technology Department

D. Laurie
Department of Mathematics
Vaal Triangle Campus of the PU for CHE
PO Box 1174
Vanderbijlpark, 1900

D. Lubinsky
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits, 2050

R. McLeod
Saltire Software Inc.
Tigard
Oregon
U.S.A

H.J. Messerschidt
Department of Computer Science and
Informatics
University of the Orange Free State
PO Box 339
Bloemfontein, 9300

M. Mphahlele
Department of Computer Science
University of the North
Private Bag X1106
Sovenga, 0727

G.D. Oosthuizen
Department of Computer Science
University of Pretoria
Pretoria, 0002

J. Owen
University of Cape Town
Rondebosch, 7701

D. Petkov
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

O. Petkova
Technikon Natal
PO Box 101112
Scotsville, 3209

N. Pillay
Department of Financial Studies
Technikon Natal, Pietermaritzburg
PO Box 101112
Scotsville, 3209

L. Pluym
Katholieke Universiteit Leuven
Celestijnenlaan 200A
B-3001 Heverlee
Belgium

K. Prag
Department of Electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

P. Premjeeth
Department of Electrical Engineering
University of Durban-Westville
Private Bag X54001
Durban, 4000

V. Ram
Department of Computer Science
University of Natal
Private Bag X01
Scotsville, 3209

J. Robertson
Department of Computer Science and
Informatics
University of the Orange Free State
PO Box 339
Bloemfontein, 9300

S. Rock
Department of Artificial Intelligence
Edinburgh University
United Kingdom

J. Roos
Department of Computer Science
University of Pretoria
Pretoria, 0002

I. Sanders
Department of Computer Science
University of the Witwatersrand
Private Bag 3
Wits, 2050
Abstract

Ease of retrieval is of crucial importance in case based reasoning (CBR). In fact, the performance of a CBR system depends on how such a system is able to retrieve relevant cases from memory and adapts them to suit new problem cases. This paper suggests a method of enhancing the retrieval process of relevant cases from memory by employing a lattice, which has potentially good indexing capabilities. It will be shown that the method is well suited for Example Based Machine Translation (EBMT), especially when the input is grammatically incorrect.

1.0 Motivation

Nagao observed that initially when a language is being learnt, lots of corresponding sentences, in the source and target languages, are memorised and as part of the learning process, a learner compares what s/he perceives with what is already memorised [Nagao, 1984].

Moreover, when a language has been mastered, a learner becomes aware of incorrect forms of the sentences in that language. Based on what has been learnt, a good guess can always be made of a correct form of a sentences and hence correct meaning inferred. This ability of humans to ‘auto-correct’ utterance makes it possible for communication to take place, even with grammatically incorrect sentences or ill-formed language.

Theories are being formulated in an attempt to model human intelligence in mastering natural language. Nagao's observation let to widely acclaimed notion of Example Based Machine Translation (EBMT), and [Nagao,1992] attempts to address a broader question of taking all words and their semantic roles into consideration when interpreting sentences in natural language, which is what humans do. Unfortunately, the machines that are used to process natural language can only see one word at a time hence it is difficult to bring other words into picture at the same time, although sophisticated parsing algorithms have been devised to bring syntactic and semantic information together. But regardless of their complexity, those algorithms are still far from mimicking human intelligence in mastering natural language.

The aim of this paper is twofold: first is to show that the use of a lattice facilitates a quick access to similar cases, since the power of an EBMT system lies in its ability to
identify relevant cases. Second is to show that the use of a lattice addresses the question by [Nagao, 1992], that of bringing into consideration all the words and their semantic roles when a similar case is being sought. That is, the use of a lattice in natural language processing is an attempt to provide an answer to how to look at a word in relation to others when interpreting a sentence, which is closer to what may be taking place in a human mind.

The cases in this approach are the English sentences that have been seen or heard and which are stored in a case base. This is the training set which will be incremented each time an unknown sentence is perceived. These cases are treated as entities by our system so that a lattice can be used to index them. Since entities are characterised by their attributes, the words form attributes of the sentences (entities) in which those words have appeared. Using a lattice as an indexing mechanism, however, disturbs the order of words in the sentences, therefore to preserve the order and to avoid duplication (of words appearing as attributes in the lattice as well as in the actual sentences), the sentences are stored as vectors called word type vectors (WTV's). More will be said about WTV's in section 3.

The structure for the rest of this paper is as follows: section 2 describes the concept of adaptation; section 3 introduces lattices and how they are used to index cases in a case base. Section 4 describes how, with the help of a lattice, grammatically incorrect sentences can be adapted and mapped onto their correct counterparts, based first on their semantic similarity and/or syntactic similarity. Section 5 gives concluding remarks and suggests a way forward.

2.0 Concept of adaptation

The effectiveness of CBR systems, and in particular Example-Based MT (EBMT) systems, depends on the richness of the accumulated and aligned source and target example pairs (case base) as well as the effectiveness of the matching algorithms [Nagao, 1992]. But it takes a long time for that rich case base to be built hence EBMT systems 'take a long time to be usefully effective' [Nirenburg et al, 1995]. As a result, cases have to be adapted to suit new problem cases.

In EBMT, cases are retrieved on the basis of their similarity to the input case. The retrieved cases are regarded as best approximations of the case at hand. The best retrieved case may not be exactly similar to the input case, but it may be adapted, that is, changed in such a way that it becomes the best translation of the input. This is necessary because of the 'sparse data problem in EBMT' [Collins et al, 1996].

The concept of adaptation is intuitively appealing since not all similar cases can be used to solve new problems, but only those cases that can be safely adapted to suit the new problem case ([Smyth et al, 1995] and [Collins et al, 1996]). This realisation became evident when CBR systems were applied to planning tasks. Adaptation knowledge is needed in order to transform and adapt retrieved cases to suit the new problem case, e.g. [Smyth et al, 1995].

As mentioned above, this paper will only concentrate on the identification of the best relevant cases. The objective is to show that if the best similar case can be retrieved
then the best translation can be attained. This translation is based on how the original sentence was translated from the source to the target. The system therefore uses the experience gained previously to solve similar problems at a later stage. The paper does not however address the transition from retrieval of the best case to the generation of the best translations. It discusses how the best cases can be retrieved and that is discussed in the next section.

3.0 A lattice and indexing

Current EBMT systems e.g. PANGLOSS [Nirenburg et al, 1995] compare input with stored strings of words and the string that contains most of the words, in an order that is as close as possible, to the input is taken to be the best approximation of the input. The ReVerb system of [Collins et al, 1996] uses both string matches and syntactic function matches, when string matches fail, in order that greater unification flexibility can be provided.

In this approach, a lattice is proposed as the indexing tool for the case base. A lattice is defined by [Oosthuizen, 1994] as a ‘directed acyclic graph with an additional constraint that every pair of nodes in the graph has a unique nearest common descendant called a ‘meet’ and a unique nearest common ancestor which is called a ‘join’. The lattices being used here are a special kind, called concept lattices, which have an additional property that apart from the nodes adjacent to the universal node T at the top and the NULL node at the bottom, no nodes in the graph have exactly one parent or exactly one child. No node has a parent (i.e. no node is directly linked to another node) to which it is also indirectly linked by means of a path that goes via one or more other nodes'.

The nodes at the top will be pointers to actual words that have appeared in the sentences in the training set. Only unique words are stored. This enables statistics on the usage of any word in a particular sense to be accumulated. For instance, it can be calculated in how many sentences a particular word has been used as a noun as opposed to its usage as a verb. The nodes at the bottom will be pointers to the actual sentences that are in the training set. Each sentence is stored in the form of a vector that contains only types of words that have appeared in that sentence. This vector will be termed word type vector (WTV). Again only unique WTV’s are stored since some sentences share structural similarities, though words may be different. Reference to common WTV’s is kept up to date so that the probability of a WTV can be determined. These accumulated statistics help in disambiguation, that is, deciding which word sense to use and which sentence structure to adopt. Figure 1 shows a sentence lattice (WTV’s and corresponding sentences are shown). To conform to CBR paradigm, there is no lexicon other than that which is stored in the lattice.

The lattice has the capability of linking every word to a WTV of a sentence in which that word has appeared. Also, all the words that tend to appear together are grouped. Starting from words (semantic functions), we are led to the WTV’s (syntactic functions). This enables both the semantic and the syntactic functions to be considered when selecting a best case. By taking the meet of all the words in the input, we get a
WTV that is linked to all of them. This WTV represents the syntactic form of the best case. Words in an input often appear together for the first time and as such there exists no meet for them (other than a new WTV they will be connected to when inserted into the case base). But examining the internal nodes of a lattice, it will be discovered that there exists a meet for the majority of the words in the input (if they are not new words in the lattice). This node will lead to a WTV that may have same syntactic features as the words in the input that did not have a meet with the identified words.

Because of the indexing mechanisms in place within a lattice, the whole case base need not be wholly explored when searching for adaptable cases, but the system is directed to relevant cases directly. The overhead affecting the system is only due to selecting the most relevant case, that is, a WTV that contains most of the words in the input. This is a good characteristic of an Adaptation Guided Retrieval (AGR) system [Smyth et al, 1995] as non adaptable cases are not examined.

The use of the lattice allows us not only access to the sentence that contains most of the words in the input, but also the correct structure of that sentence in the form of a WTV. Even if such an input sentence contains some grammatical mistakes, indexing capabilities of the lattice enables it to be mapped onto a closest similar sentence. The fact that a grammatically incorrect sentence can be mapped onto a closest similar sentence suggests that the use of a lattice can be extended to parse grammatically incorrect sentences. This parsing means attempting to map grammatically incorrect sentences onto their correct forms or counterparts, as a sentence may have seen before. The next section discusses how grammatically incorrect input can be adapted in a lattice indexed case base.

4.0 Filling knowledge gaps - A CBR Approach

Knowledge gaps are a concern and a hindrance to smooth operation of MT systems. Some MT systems (e.g. [Winiwater et al, 1996]; [Knight et al, 1995]; [Dorr et al, 1995]; [Yamada et al, 1995]), are built with parsers equipped for handling ungrammatical input. A system can be designed to respond to a knowledge gap in various ways. A system may '...signal an error, ...make a random decision, ...appeal to human operator or ...turn to robust MT programs' [Knight et al, 1995], for help. The JANUS system [Mayfield et al, 1995], assumes that, though spontaneous speech may be ill-formed, it is semantically correct, and therefore the system concentrates on and regulates the semantics at the phrase level, thus ignoring syntactic category of the input string. This means that unknown/misspelled words, omitted words and/or redundant words are ignored. The aim of the JANUS project is not 'translation in the traditional sense, but rather producing an equivalent message in the target language based on the meaning' [Mayfield et al, 1995]. The question is what: happens when the ignored syntactic category hinders the inference of the meaning?

The major advantage of using a lattice is to enhance the robustness of MT programs in dealing with knowledge gaps, especially in EBMT. Using a lattice allows access to two pieces of information concurrently: we get matching words and sentence structures. If a similar case is not retrieved based on similar words (semantic similarity), it is retrieved based on the sentence structure, to which it can be adapted. Besides lattices being able to index words to the sentences in which those words have
Initially been used, they are capable of accumulating statistics by recording how many references have been made to one WTV or how many times a word has been used in a particular sense. The ability to accumulate these statistics make lattices suitable for NLP applications.

As mentioned above, a lattice does not only allow access to stored phrases, but the sentence structure (WTV) in which those phrases have occurred. It can be observed that the WTV's are nothing but extracted parse subtrees of a broader parse tree of a language in question, in this case English. Each WTV therefore corresponds to a rule in a rule based MT system. If one comes across the following phrase, for instance:

... planes flying ahead. - (1)

and later comes across the phrase:

... birds flying ahead. - (2)

With the lattice, we will be able to retrieve the WTV of a sentence in (1), in addition to others to be retrieved depending on the words that preceded the quoted phrase. This is because a phrase structure in (2) is the same as a phrase structure in (1) apart from the fact that the two contain 'flying ahead' in common. That similar structure is:

<...noun,verb,adverb>.

The WTV's that are retrieved first have words in common (string based matching), then similar (syntactic) structure with an input string. While other EBMT systems may claim to be doing the same thing, the difference between them and the lattice approach is that we do not get semantic matches in isolation to syntactic matches, (as shown in the example above). The indexing mechanisms of a lattice also gives closest neighbours (to the input) ranked according to how many attributes each neighbour has. A neighbour with most of the desired attributes ranks the top while one with least number of attributes ranks the bottom.

In conventional EBMT, one gets chunks of phrases which need some joining and figuring out how the translation is to be configured. In lattice-based natural language processing, one gets the sentence structure (WTV’s) in which those phrases have appeared and in addition to the obtained phrases. Each WTV's has been shown to be equivalent to a rule in a Rule - Based MT system. With a little luck, either the whole input sentence could be the same as the stored one or if the words are different their syntactic roles could be the same as those that have been used in the construction of the original WTV. Thus resulting in a WTV similar to the retrieved one. This being the case, the experience gained in translating the original sentence can be used to generate the translation for the input sentence. In Rule Based MT systems, a general rule is designed and all the input matching that rule, is translated into the target language using a single rule designed for that purpose.

As mentioned above, when a grammatically incorrect utterance is perceived, humans are capable of deriving the true sense of the utterance by comparing it to some similar utterance heard some time back (adapting it). The similarity can either be based on
similar words or similar sentence structure (WTV's). This implies that the utterances that are being heard are stored (vocabulary and language knowledge) for future reference, hence why communication is possible even if input can be grammatically incorrect. Words in a lattice form a vocabulary, WTV's form correct sentence structures (equivalent to language rules) and links of a lattice connect words to their correct sentence structures (WTV's). If words that have not appeared in any WTV (sentence structure) are conceived for the first time, then their sentence structure can be adapted, and if such an utterance is grammatically incorrect, it can be labelled as such based on the results of its comparison to a WTV it is closest to. The current research aims to deal with grammatically incorrect sentences. These sentences could be input to a machine translation system, a database front end, a speech to text generation system (since speech, like written text, is inherently riddled by mistakes) etc.

Many mistakes appear in writing and these often lead to failure of input sentences to parse and subsequent failure for an NLP program to execute. These mistakes that appear in writing are inevitable since 'written text make a very large portion of communication' [Goldfain, 1995].

Grammatical mistakes that can occur in a sentence can be categorised into sentences with misspelled word(s), sentences with (an) omitted word(s), sentences with repeated/redundant word(s), and sentences with incorrect word order. A sentence first has to be parsed before it can be labelled as containing a grammatical mistake, and this happens when parsing fails. When lattice is used to parse a sentence, a corresponding WTV is generated for that sentence. The generated WTV is then matched against the existing WTV's and when it fails to match, then parsing has failed. The sentence is then suspected to fall into one of the above mentioned categories. The next section explains how sentences are treated in each of the mentioned categories. The examples used here are simple ones, based on the lattice in figure 1, but the actual situation can be much more complex than that presented in the quoted examples.

4.1 Sentence with misspelled words

A misspelled word cannot be recognised by the parser which assigns words their semantic roles as they are found in the lattice. Because such a word cannot be recognised a WTV for a sentence with a misspelled word will have a 'U' in the place of an unrecognised word's type. For simplicity, only WTV's with one unknown word will be visited. The closest WTV is searched for by taking the meet of the words contained in the input. With a little luck, one WTV may be found in the case base which is different in one position from the WTV for the input sentence. A new WTV is then generated by substituting a 'U' in the input WTV by a different position in the retrieved WTV. The new WTV then becomes the best approximation of the input.

Two possibilities then arise: Firstly, the input could have been seen before and that means the misspelled word is already in the lattice and could be retrieved and substituted safely. Secondly, a misspelled word could have the same syntactic role which can make the WTV for the input the same as the retrieved WTV. This can be realised if a misspelled word is subjected to a spelling correction algorithm for correction.
To illustrate, let us consider the following example of a sentence with one misspelled word.

‘we love the surroundings’

where ‘surroundings’ is misspelled.

The resulting WTV for the sentence is (after using the lattice in figure 1):

\[ G = \langle pp, v, art, U \rangle \]

the meet of successfully located words leads to \( S_3 \) and \( S_3 \) has the same length as \( G \) and it differs from it in only one position. Thus:

\[
S_3 = \langle pp, v, art, n \rangle \\
G = \langle pp, v, art, U \rangle
\]

We therefore conclude that \( G \) is closest to \( S_3 \) and therefore expect the spelling auto correction to yield a ‘noun’ which will make \( G \) same as \( S_3 \).

This ability of a lattice to give probable expected types make the lattice an attractive tool to use. String-based comparisons cannot shed this kind of light on the incorrect input sentence.

4.2 Sentence with an omitted word

Omitted words can sometimes be as harmless as an article or as crucial as an important term without which it would be difficult to make out the meaning of a sentence. A WTV of a sentence with an omitted word will have at least one type less than its correct equivalent. The located meet of the words in the input will lead to a set of WTV’s to be retrieved. Again, we restrict the discussion to sentences with one omitted word. A closer examination of the selected set of WTV’s will result in at least one WTV which is one word type longer than the WTV corresponding to the input sentence (generated WTV). Having discovered one such WTV, a missing word type is inserted into the corresponding slot in the generated WTV and the resulting (target) WTV becomes the best approximation of the input. If a correct version of a sentence has previously been built into the lattice, then the meet of available words will lead to the desired WTV and hence the omitted word would be recovered.

To illustrate, consider the following sentence:

‘we the surroundings’

where ‘love’ has been omitted.

Taking the meets of available words using the lattice in figure 1, we generate the following WTV which is at least one type shorter than \( S_3 \)

\[
G = \langle pp, art, n \rangle \\
S_3 = \langle pp, v, art, n \rangle
\]
whereby every type that is present in G is also in S₃, and the meet of the words in the input lead to S₃. We therefore conclude that G must be closest to S₃.

From observation, G is probably lacking a verb to make it the same as S₃. The system may therefore issue a warning of a required or lacking verb.

### 4.3 Sentence with a redundant/repeated word

Redundant words are common in written text. A WTV of a sentence with a repeated or redundant word is a variation of the one with an omitted word, discussed above. It will have one of its types repeated, probably consecutively as happens regularly in written text. Such a WTV will therefore be at least one type longer than its correct counterpart. For this phenomenon, a WTV can be treated as a set since this will allow a match to occur even if one WTV has a repeated member, that is, a word type. A search for a WTV that contains the same elements as the generated WTV (corresponding to the input sentence) is conducted similarly by taking the meet of the words in the input. Having found the retrieved WTV, the generated (incorrect) WTV is made the same as the correct one by deleting the redundant word type hence the redundant word in the input.

To illustrate, consider the following sentence:

'we love the the surroundings'

where 'the' is a redundant word, and thus results in the following generated WTV, which is at least one type longer than S₃, with every type that is present in G also in S₃:

\[
G = <\text{pp, v, art, art, n}>
\]

\[
S₃ = <\text{pp, v, art, n}>
\]

Also the meet of the words in the input leads to S₃. We therefore conclude that G is closest to S₃. If G and S₃ are treated as sets, then they will be found to be equal. G is probably having a redundant article without which it will be the same as S₃, which is taken to be the best approximation of the input. The redundant article in the input will therefore be deleted to make G the same as S₃.

### 4.4 Incorrect word order in a sentence

An unrestricted text is a free domain and that freedom often brings inconsistencies. Some types such as an adverb in the English language can be shifted across the sentence without changing the meaning of the input. For instance, consider the following sentences in which a stated adverb can be placed at either one of the blanks, indicated by [], in the sentences:

1. quickly
   She [] ate her food [].

2. next
   [] we are going to explain how the sentences in each category are treated []
3. even  

The organisation cannot establish a centre for the homeless.

Rules in an MT system, for instance, are written in anticipation of a certain formation of word types and, in particular, Lexica\(^2\) assumes words in an input stream to follow a particular sequence failing which the parse will not be successful. Variations of the same rule have to be written to cater for different formations of sentence. This can be difficult especially in unrestricted text domains.

With the use of a lattice, we need only store one variation of the sentence and the rest of the input will be changed in accordance with the stored sentence structure. The words can then be re-arranged according to the correct WTV and the sentence can then be submitted for further processing without problems. With this approach, rule based MT developers will not need to have many rules, each catering for a variation of the same utterance. With just one rule and relying on the capabilities of a lattice to transform any variation and map it onto the form that has been catered for, MT engines can operate without hindrance from errors emanating as a result of incorrect word order in an input sentence.

5.0 Conclusion and future work

The proposed case retrieval method combines the structural (syntactic) form, the semantic (words) form of the sentences and the dynamic indexing mechanisms between the words and the WTV's (structural form of the sentences). The method does not look for similar words in isolation to similar sentence structure, but combines the two in a manner no approach has mastered. This brings in the semantic and structural environments together, unlike other approaches which locate either similar words [Nirenburg et al, 1995] or partial structures [Collins et al, 1996].

Also when it comes to handling of a knowledge gap, it is done so in relation to other words (syntactic/semantic functions) in the sentence. This demonstrates that a lattice is able to bring together all other words in the sentence when interpreting a sentence.

In future the project will be extended to detection of noun - verb agreement in English language. This will form the basis of automatic correction of sentences, say, in a word processing session. This can also be a good tool to aid people learning English to write grammatically, that is syntactically, correct English sentences.

Unique words are stored in a lattice and yet a word can play many roles in various sentences. A word can be used as a noun in one sentence and as a verb in another sentence. The use of a lattice allows one to calculate the number of instances a particular word been used, say, as a noun as opposed to its usage as a verb. Also, as a particular WTV (sentence structure) keeps recurring in the case base, the number of references are recorded and can later serve as an additional statistical tool. These statistics could be used to disambiguate words or sentence structures when need be. The advantage of this is evident: being able to choose the right sentence structure puts

\(^2\)Lexica is an MT engine developed at the University of Pretoria. It is a rule-based lexical transfer MT engine supporting translation between two natural language in an unrestricted domain.
all the words onto their correct parts of speech, hence the words are disambiguated as well.

And lastly, the author believe that the ability of lattices to combine CBR and in particular, EBMT, with statistics, can be expected to be of significance importance in NLP systems.
Success is all that is looked for.
We love the surroundings.
We looked around heaven all day.
Love is all around us.
She looked around the corner to see a dog.
Around is all the surroundings around us.
Success is all that is looked for.
where:
n = noun
aux = auxiliary
cop = copulative
adv = adverb
pp = personal pronoun (i.e. pronouns that refer to people only)
adj = adjective
inf = infinitive
v = verb
op = object pronoun (i.e. pronouns that refer to things/animals only)
art = article
prep = preposition
6.0 References


2. [Collins et al, 1996]; Collins B., and Cunningham P.; Adaptation-Guided Retrieval in EBMT: A Case Based Approach to Machine Translation. In Ian Smith and Boi Falting (Eds): Advances in Case-Based Reasoning; Proceedings of the Third European Workshop on Case-Based Reasoning; Lausanne, Switzerland; November 1996; Lecture notes in Artificial Intelligence, Vol 1168; Springer, pages 91 - 104.


5. [Goldfain, 1995]; Goldfain M. S.; Representing Space and Time information in English Narratives; Knowledge Representation for Natural Language Processing in Implemented System; American Association for Artificial Intelligence Press, 1995; Menlo Park, USA; pages 45 - 53.


11. [Nagao, 1992]; Nagao M.; Are the grammars so far developed appropriate to recognize the real structure of a sentence? TMI-92 Fourth International Conference on Theoretical and Methodological Issues in Machine Translation, June 1992; Montreal Canada. pages 127-137.


19. [Smyth et al, 1995]; Smyth B., and Keane M.; Experiments On Adaptation-Guided Retrieval In Case Based Design; In Veloso Manuel (Ed): Case - Based Reasoning -
Research and Development; Proceedings of the First International Conference on Case Based Reasoning, ICCBR-95; Sesimbra, Portugal, October 1995; Lecture notes in Artificial Intelligence; Springer, pages 313 - 324.
