Volume 3 ▪ Issue 1 ▪ October 2012

# INTERNATIONAL JOURNAL OF COMPUTATIONAL LINGUISTICS (IJCL)

Publication Frequency: 6 Issues / Year

ISSN : 2180-1266

CSC PUBLISHERS

http://www.cscjournals.org

# INTERNATIONAL JOURNAL OF COMPUTATIONAL LINGUISTICS (IJCL)

**VOLUME 3, ISSUE 1, 2012**

**EDITED BY**
**DR. NABEEL TAHIR**

ISSN (Online): 2180 - 1266

International Journal of Computational Linguistics (IJCL) is published both in traditional paper form and in Internet. This journal is published at the website http://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJCL Journal is a part of CSC Publishers

Computer Science Journals

http://www.cscjournals.org

# INTERNATIONAL JOURNAL OF COMPUTATIONAL LINGUISTICS (IJCL)

Book: Volume 3, Issue 1, October 2012
Publishing Date: 31-12-2012
ISSN (Online): 2180-1266



IJCL Journal is a part of CSC Publishers
http://www.cscjournals.org


Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

**CSC Publishers, 2012**

# EDITORIAL PREFACE

The International Journal of Computational Linguistics (IJCL) is an effective medium for interchange of high quality theoretical and applied research in Computational Linguistics from theoretical research to application development. This is the *First* Issue of Volume *Three* of IJCL. The Journal is published bi-monthly, with papers being peer reviewed to high international standards. International Journal of Computational Linguistics (IJCL) publish papers that describe state of the art techniques, scientific research studies and results in computational linguistics in general but on theoretical linguistics, psycholinguistics, natural language processing, grammatical inference, machine learning and cognitive science computational models of linguistic theorizing: standard and enriched context free models, principles and parameters models, optimality theory and researchers working within the minimalist program, and other approaches.

IJCL give an opportunity to scientists, researchers, and vendors from different disciplines of Artificial Intelligence to share the ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development. This journal is helpful for the researchers and R&D engineers, scientists all those persons who are involve in Computational Linguistics.

Highly professional scholars give their efforts, valuable time, expertise and motivation to IJCL as Editorial board members. All submissions are evaluated by the International Editorial Board. The International Editorial Board ensures that significant developments in image processing from around the world are reflected in the IJCL publications.

IJCL editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Scribd, CiteSeerX Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCL. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJCL provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

**Editorial Board Members**
International Journal of Computational Linguistics (IJCL)

# EDITORIAL BOARD

# TABLE OF CONTENTS

Volume 3, Issue 1, October 2012

## Pages

# Design and Development of a Malayalam to English Translator- A Transfer Based Approach

**Latha R Nair** *latha5074@gmail.com*
*Assistant Professor*
*School of Engineering*
*Cochin University of Science and Technology*
*Kochi,Kerala,682022, India*

**David Peter S** *davidpeter@cusat.ac.in*
*Professor*
*School of Engineering*
*Cochin University of Science and Technology*
*Kochi,Kerala,682022,India*

**Renjith P Ravindran**
*School of Engineering* *renjithforever@gmail.com*
*Cochin University of Science and Technology*
*Kochi,Kerala,682022,India*

**Abstract**

This paper describes a transfer based scheme for translating Malayalam, a Dravidian language, to English. The input to the system is a Malayalam sentence and he ouput isits equivalent English sentence. The system comprises of a preprocessor for splitting the compound words, a morphological parser for context disambiguation and chunking, a syntactic structure transfer module and a bilingual dictionary. All the modules are morpheme based to reduce dictionary size. The system does not rely on a stochastic approach and it is based on a rule-based architecture along with various linguistic knowledge components of both Malayalam and English. The system uses two sets of rules: rules for Malayalam morphology and rules for syntactic structure transfer from Malayalam to English. The system is designed using artificial intelligence techniques and can easily be modified to build translation systems for other language pairs.

**Keywords:** Malayalam Language, Transfer Based Approach, Machine Translation, Morphological Parser.

## 1. INTRODUCTION

Work in the area of Machine translation in India has been going on for several decades. Promising translation technology began to emerge by 1970 with the developments in the field of artificial intelligence and computational linguistics. Machine Translation Systems in certain well-defined domains have been successfully developed. Translation of gazette notifications, office memorandums,and circulars has been done successfully by Mantra system developed by centre for development for advanced computing(CDAC), Pune. Most of the systems developed are for Hindi, the official language of India. This paper describes a translator for translating sentences in Malayalam a Dravidian Language to English developed on a rule based architecture combined with linguistic knowledge components of both Malayalam and English. The system has a preprocessor for splitting the compound words, morphological parser for context disambiguation and chunking and a bilingual dictionary. A set of rules for Malayalam morphology and rules for syntactic structure transfer from Malayalam to English have been incorporated in the system.

Some of the organizations which are involved in the development of translation systems are: Indian Institute of Technology (Kanpur), Center for Development of Advanced Computing (CDAC) (Mumbai), CDAC(Pune), Indian Institute of Information Technology (Hyderabad) . They are

engaged in development of MT systems under projects sponsored by Department of Electronics, state governments etc. since 1990[1,2]. Research on MT systems between Indian and foreign languages and also between Indian languages are going on in these institutions.

The two major goals in any translation system development wrk are accuracy of translation and speed. Accuracy-wise, smart tools for handling transfer grammar and translation standards including equivalent words, expressions, phrases and styles in the target language are to be developed. The grammar should be optimized with a view to obtaining a single correct parse and hence a single translated output. Speed-wise, innovative use of corpus analysis, efficient parsing algorithm, design of efficient Data Structure and run-time frequency-based rearrangement of the grammar which substantially reduces the parsing and generation time are required [3]. A fully automatic Machine translation system should have different modules such as morphological analyzer, Part of speech tagger, chunker, Named entity recognizer, word sense disambiguator, syntactic transfer module and target word generator [3]. The different techniques used for translation differs in the number of modules used and also the way these modules are implemented. Both rule based and statistical approaches have been tried in the implementation of each of these modules.

The various approaches used in the MT systems for Indian languages are: Direct machine translation systems, Rule based systems and Corpus based systems. Rule based systems do not use any intermediate representation. This is done on a word by word translation using a bilingual dictionary usually followed by some syntactic arrangement. [4, 5,6] 2) Rule based translation which produces an intermediate representation, which may be a parse tree or some abstract representation. The target language text is generated from the intermediate representation. Of the two rule based methods, Interlingua and transfer based approach, transfer based systems are more flexible and it can be extended to language pairs in a multilingual environment. The Interlingua based systems can be used for multilingual translation [7]. The amount of analysis needed in Interlingua approach is more than that in a transfer based approach. The universal networking language has been proposed as the Interlingua by the United Nations University for overcoming the language barrier[8]. Corpus based MT is fully automatic and requires less human labour than rule based approaches. The disadvantage is that they need sentence aligned parallel text for each language pair and this method can not be employed where these corpora are not available [9, 10].

## 2. PREVIOUS WORK

English to Hindi MT system Mantra, developed by Applied Artificial Intelligence (AAI) group of CDAC, Bangalore, in 1999 uses transfer based approach. The system translates domain specific documents in the field of personal administration; specifically gazette notifications, office orders, office memorandums and circulars. It is based on lexicalized tree adjoining grammar (LTAG) to represent English and Hindi grammar which are used to parse source English sentences and for structural transfer from English to Hindi [2]. This system also works well on other language pairs such as English-Bengali, English-Telugu, English-Gujarati , Hindi-English etc and also between Indian language pairs such as Hindi-Bengali and Hindi-Punjabi. The Mantra approach is general but the lexicon and grammar have been limited to the specific domain of personal Administration. It uses preprocessing tools like phrase marker, named entity recognizer, spell and grammatical checker. It uses Earley's style bottom up parsing algorithm for parsing. The system provides online addition of grammar rule. The system produces multiple translation results in the case of multiple correct parses.

English to Kannada MT system has been developed at Resource centre for Indian Language Technology Solutions (RC_ILTS), University of Hyderabad by Dr. K. Narayan Murthy [2]. This also uses a transfer based approach and it can be applied to the domain of government circulars. The project is funded by Karnataka government. This system uses Universal Clause Structure Grammar (UCSG) formalism [15]. The technique is applied to English_ Telugu translation as well.

Other systems developed using this approach are : Matra- English to Hindi MTS developed by CDAC, Pune, Sakti- English to Marathi, Hindi and Telugu developed by IISc Bangalore and IIIT Hyderabad, Anubaad- English to Bengali developed by CDAC, Kolkata, English to Malayalam MTS developed by Amrita Institute of Technology.

It is found that translation between structurally similar languages like Hindi and Punjabi can be developed easily than translation systems between Indian languages and English which differ in the syntactic structure. The proposed translation system translates Malayalam sentences to English sentences. Since there is a wide difference in English and Malayalam sentences the system needs an additional modules for parsing and syntactic reordering.

## 3. DEVELOPMENT AND IMPLEMENTATION OF TRANSFER BASED MACHINE TRANSLATION SYSTEM

A transfer based MT system has been developed with the following system modules 1. A preprocessor for splitting the compound words [13] 2. a morphological parser for context disambiguation and chunking 3. A transfer module which transfers the source language structure representation to a target language representation. 4. A generation module which generates target language text using target language structure. Block diagram of the same is shown in fig 1. The grammar rules for Malayalam and some of the transfer rules for transferring source parse tree to target parse tree are stored in two separate files. Some of the transfer rules are embedded in the source code. The sentences stored in a source file are read one by one by the input module and given to the preprocessor module. The final translated output is stored in another file.

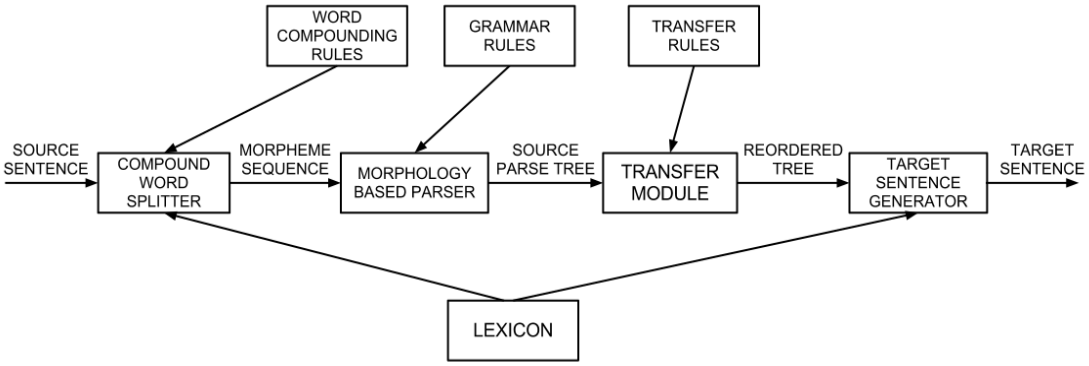



**FIGURE 1:** Block diagram of a transfer based system

### 3.1 Compound Word Splitter Module

Morphological variations for words occur in Malayalam due to inflections, derivations and word compounding. Malayalam is an agglutinative language where words of different syntactic categories are combined to form a single word. Formation of new words by combining a noun and a noun, noun and adjective, verb and noun, adverb and verb, adjective and noun and in some cases all the words of an entire sentence to reflect the semantics of the sentence are very common. The complexity of compounding in Malayalam language can be understood from the following example.

സീതയുടെപൂച്ചയൊരെലിയെത്തിന്നു- (1)

The English version being Seetha's cat ate a rat

The constituent words in 1 are to be separated before any further processing. Splitting has been done at morpheme level to reduce dictionary space. The above sentence gets split as shown in 2

സീത ഉടെ പൂച്ച ഒരു എലി എ തിന്നു--(2)

morpheme by morpheme translation for the sentence at 2 is :

Seetha 's cat a mouse (null) ate

The morpheme sequence will be as in 2 above. The sequence of morphemes is given to the parser for chunking and word sense disambiguation. The set of inflectional suffixes for nouns and verbs and derivational suffix for adjectives are based on previous works [11, 12]. Due to the ambiguity in the splitting rules the system generates multiple splits for the same input sentence and the split with least number of constituents is fed to parser.

**3.2 Parser Module**

Parser takes input from the splitter and does the following tasks. It groups the input sequence of morphemes into chunks [14, 15] and performs word sense disambiguation based on morpheme tags [16]. The chunking process finds the basic units for tree reordering. The word sense disambiguation is required as a morpheme can have multiple tags. The parser uses a depth first approach with backtracking [17]. The output of the parser is a parse tree for the next module. The parser uses the syntax rules for the morpheme sequences in Malayalam sentences in the regular expression form. A set syntax rules in the regular expression form are shown below:

1. S-> NP*VP
2. NP-> ADJ*NP | N NA
3. VP ->ADV* V VA| V VA

Rule 1 implies that a simple sentence is a sequence of noun chunks followed by a verb chunk. Based on the second rule, a noun chunk consists of a set of adjectives followed by a noun and suffixes like case, gender and number for nouns. According to the third rule a verb chunk consist of a sequence of adverbs followed by a verb. Only a subset of such rules derived is shown above. The chunks selected form groups for structural transfer to form target language structure.

A sample sentence and the parse tree generated for the sentence using the grammar rules are shown below:

Input sentence:

മാല മോഷ്ടിച്ച കള്ളന്മാർ രാത്രിയിൽ കാട്ടിലേക്ക് പോയെന്ന് പോലീസ് വിചാരിച്ചു.

English version: The police thought that the thieves who stole the chain went into forest in the night.

Output of the splitter:

മാല മോഷ്ടിച്ച കള്ളൻ മാർ രാത്രി ഇല് കാട് ലേക്ക് പോയി എന്ന് പോലീസ് വിചാരിച്ച

English version:chain stole theif 's night in forest to went that police thought

Output of the parser:

CS(NC(S(NG(ADJC(S(N (മാല ) V(മോഷ്ടിച്ച) RP) NG(N(കള്ളൻ) PL(മാർ))) NG(N(രാത്രി ) NA(ഇൽ)) NG(N(കാട്)

NA(ലേക്ക്)) V(പോയി )) NCA(എന്ന്)) S(N(പോലീസ്) V(വിചാരിച്ച)))

English version: CS(NC(S(NG(ADJC(S(N (chain ) V(stole) RP) NG(N(theif) PL('s))) NG(N(night) NA(in)) NG(N(forest) NA(to)) V(went )) NCA(that)) S(N(police) V(thought)))

The corresponding parse tree generated is shown in Fig.2

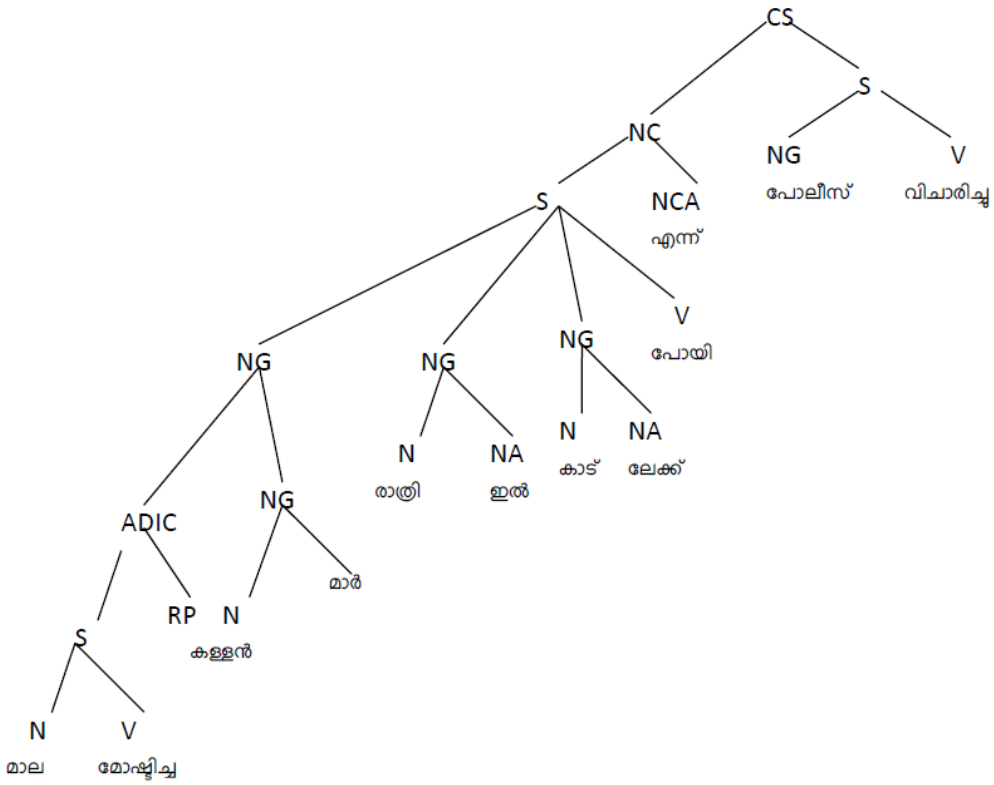
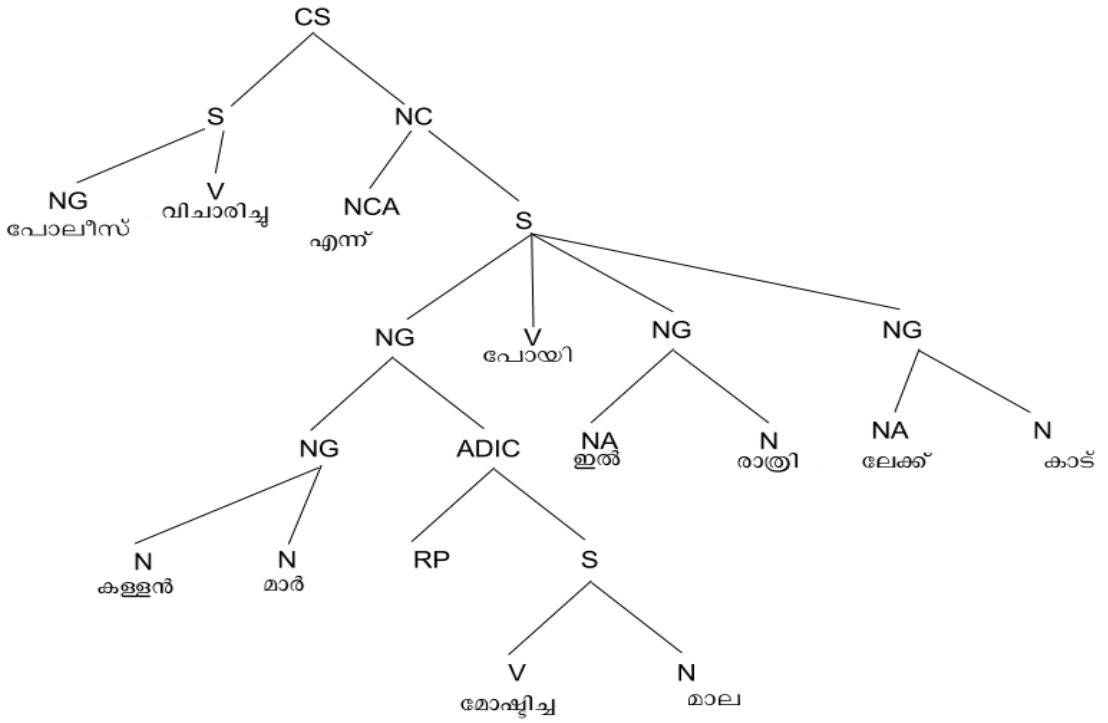


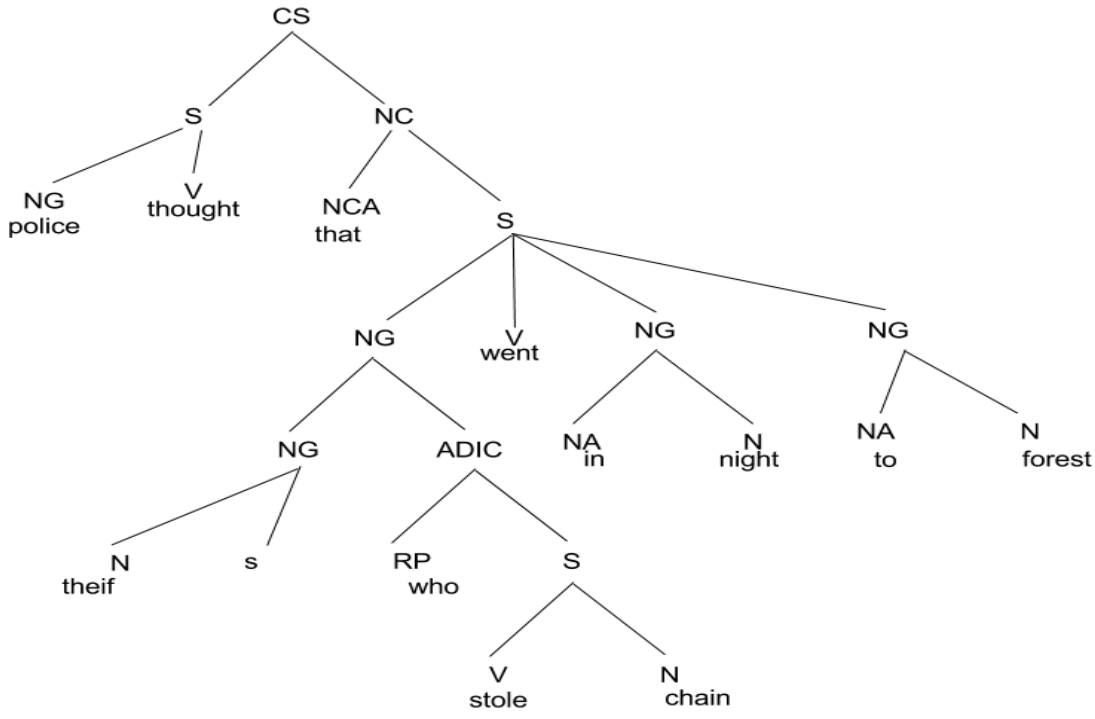

**FIGURE 2:** Generated Parse tree

### 3.3 Syntactic Structure Transfer Module

The transfer module transfers the source language structure representation to a target language representation. This module needs the sub tree rearrangement rules by which the source




a




b

**FIGURE 3:** a. Parse tree after structural transfer b. Corresponding parse tree in English

language sentence syntax tree can be transformed into target language sentence syntax tree. The system performs most of the commonly needed reordering for Malayalam to English translation. The tree after reordering for the above Malayalam sentence using the transfer grammar rules using the transfer rule identified is shown in fig3(a) and its corresponding English tree is shown in fig3(b).A set of transfer rules used by the system are shown in Table I.

| | Malayalam structure | English structure |
|---|---|---|
| 1 | PP: NG P | PP: P NG |
| 2 | VG: ADV V | VG: V ADV |

**TABLE 1:** A set of system transfer rules

According to the first rule the order of case suffix and noun chunk should be interchanged in a prepositional chunk. The Second rule accords that in verbal chunk the adverb and verb should be interchanged.

**3.4 Target Sentence Generator Module**

The generation module generates target language text using target language structure [18]. This uses inter chunk dependency rules and intra chunk dependency rules. It involves lexical transfer of verbs, transfer of auxiliary verb for tense, aspect and mood and transfer of gender, number and person information. A depth first traversal of the target parse tree generates the following English sentence

Input Malayalam sentence:

മാല മോഷ്ടിച്ച കള്ളന്മാർ രാത്രിയിൽ കാട്ടിലേക്ക് പോയെന്ന് പോലീസ് വിചാരിച്ചു.

Correct English translation: The police thought that the thieves who stole the chain went into the forest in the night

Sentence generated by the system:

The Police thought that thieves who stole the chain went in night to forest.

**3.5 Cross lingual Dictionary**

The dictionary includes most of the commonly occurring verbs, nouns, pronouns, adjectives, inflectional and derivational suffixes, clause suffixes etc. Each entry in the file has three fields: the root word (morpheme), the morpheme tag and its translation. The verbs in past tense have their root words stored along with them. Since the system works with morphemes, the space required for the dictionary is less.

| Root word | Morpheme tag | Translation |
|---|---|---|
| പൂച്ച | Noun | cat |
| ഉടെ | Case suffix | 's |

**TABLE 2:** Lexicon

Presently the system works for sentences which contain upto two adverbial or adjectival clauses which is commonly found in Malayalam texts. The system can be modified to handle other sentences by adding appropriate grammar rules and transfer rules to the rule database. As the parser is a general parser, it can handle sentences of any depth.

### 3.6. Implementation and Testing of the System

The system was implemented in Python language and tested with a source file which contains 1000 sentences. The sentences which follow the grammar rules were translated. A group of results are tabulated in table 3.

## 4. RESULTS AND DISCUSSION

The system was tested with more than 1000 different kinds of sentences with and without subordinate clauses which follows the identified morpheme sequences. The system returned correct meaningful translations in most of the cases. A group of sample input sentences with the tabulated outputs are shown in table 3 to give a correct picture of the results obtained..In around 20% of sentences the system returned the exact English version of the input sentences. In balance translations the output sentences were meaningful but had small shortcomings due to the following reasons:

i) The positioning of articles is not considered.
ii) Many inter chunk and intra chunk dependencies are not considered.
iii) The lexicon stores only the common translation for polysemous words.

The system takes care of word sense disambiguation based on lexical category successfully. The compound nouns are also not handled by the system as the shallow parser cannot group them using the current set of rules. The system output can be enhanced including rules which can take care of the above shortcomings.

## 5. CONCLUSION

Various MT groups have used different formalisms best suited to their applications. Of them transfer based systems are more flexible and it can be extended to language pairs in a multilingual environment. A transfer based MT system has been developed for Malayalam, a Dravidian Language which comprises of a preprocessor for splitting the compound words, a morphological parser for context disambiguation and chunking, a syntactic structure transfer module and a bilingual dictionary. The system was tested successfully for more than 1000 different types of sentences wherein the system returned true results for sentences which contain two subordinate clauses. Even for sentences with more than two subordinate clauses the system

| System Input and Output | Remarks |
|---|---|
| **1. Input:** ഒരു നല്ലവനായ രാജാവ് ഒരിടത്തൊരിടത്ത് ഉണ്ടായിരുന്നു.<br>**Word to word translation:** a kind king in a place was.<br>**English version of the sentence:** There was a kind king in a place. | System output is in line with the English version except for the positioning of the article<br>.Meaningfully correct sentence |
| **System Output:** A kind king was in a place. | |
| **2.Input:**. രാജാവിന്റെ മകൾ അതിസുന്ദരി ആയിരുന്നു.<br>**Word to word translation:** King's daughter very beautiful was<br>**English version:** The king's daughter was very beautiful. | System output is exactly in line with the English version except for the positioning of article to the noun king since same word is not there in input language.<br>Meaningful correct translation |
| **System Output:** King's daughter was very beautiful. | |
| **3.Input:** ആ രാജകുമാരിക്ക് സൂര്യനെപ്പോലെ തിളങ്ങുന്ന ഒരു സ്വർണ്ണപന്തുണ്ടായിരുന്നു.<br>**Word to word translation:** That princess sun like shining is a goldenball.<br>English **version** : The princess had a golden ball which was shining like the sun. | System output gives translation without positioning of article for the nouns. The variations in translations for ആ have not been considered.<br>System output is meaningful correcttranslation |
| **System Output:** That princess had a ball which is shining like sun. | |
| **4.Input:** രാജകുമാരിക്ക് പൂന്തോട്ടത്തിൽ പന്തു കളിക്കാൻ ഇഷ്ടമായിരുന്നു.<br>**Word to word translation:** Princess garden in ball play to liked<br>**English Version** : The princess liked to play in the garden | Output gives translation without positioning of preposition since same word is not there in input language<br>Meaningful correct translation |
| **System Output:** Princess liked to play in garden. | |
| **5.Input:** രാജകുമാരി ഒരു ദിവസം കളിച്ചുകൊണ്ടിരുന്നപ്പോൾ സ്വർണ്ണപന്ത് അടുത്തുള്ള കിണറ്റിൽ വീണു.<br>**Word to word translation:** princess a day play was when golden ball nearby well in fell.<br>**English Version** : When the princess was playing one day the golden ball fell into a nearby well. | Output translation is without positioning of presposition since same word is not there in input language.<br>Multiple translations of ഒരു has not been considered.<br>Meaningful correct translation |
| **System Output:** When princess was playing a day golden ball fell in nearby well. | |
| **6.Input:** രാജകുമാരി കരയാൻ തുടങ്ങി.<br>**Word to word translation** :Princess cry to started<br>**English version** : The princess started to cry | System output gives exact English version of input sentence |
| **System Output:** Princess started to cry. | |
| **7.Input:** കിണറ്റുവക്കിലിരുന്ന ഒരു തവള ഇതെല്ലാം കാണുന്നുണ്ടായിരുന്നു<br>**Word to word translation:** well side in sat a frog all these see was<br>**English version:** A frog sitting beside the well was seeing all these. | Meaningful correct translation which slightly varies from the English version. |
| **System Output:** frog which was sitting in side of well was seeing all these. | |
| **8.Input:** തവള രാജകുമാരിയുടെ അടുത്തേക്ക് ചാടി ചെന്നു.<br>**Word to word translation:** frog princess's side jumped went<br>**English version** : The frog jumped to the princess. | Meaningful correct translation which slightly varies from the English version. |
| System Output: frog went jumping to princess's side. | |

**TABLE 3:** Group of Tabulated results

returned translated output sentences which could give basic understanding of the input sentences. More rules can be added to make the system to give exact translation of input sentences in all cases. Additional modules like finding and replacing collocations, finding and replacing named entities can also be added to the basic translator. The results obtained are encouraging. The work can be extended to create a full fledged machine translator from any Dravidian language to English since they all exhibit structural homogeneity.

## 6. REFERENCES

[1] P Dubey et al. "Overcoming the Digital Divide through Machine Translation". *TranslationJournal.,* Vol.15, 2011, http://translationjournal.net/journal/55mt_india.htm [Dec 12, 2011].

[2] B.K.Murthy, W.R Deshpande ., "Language technology in India: past, present and future", 1998,

[3] http://www.cicc.or.jp/english/hyoujyunka/mlit3/7-12.html [Dec 11,2011]

[4] S.Lalithadevi, P.Pralayankar , V.Kavitha. "Translation of Hindi se to Tamil in a MT System". Information systems for Indian languages, Berlin Heidelberg: Springer-Verlag, 2011, pp. 246–249.

[5] V Goyal, G S Lehal. "Advances in Machine Translation Systems". *Language In India,* Vol. 9, No. 11, 2009, pp. 138-150 .

**[6]** G.S.Josan , G.S. Lehal. "Evaluation of Hindi to Punjabi Machine Translation System". *International Journal of Computer Science Issues*, vol4 no1, 2009, pp 243-257.

[7] V.Goyal, G.S. Lehal . "Web Based Hindi to Punjabi Machine Translation System". *Journal of Emerging Technologies in Web Intelligence.* Vol.2., 2010, pp.148-151.

[8] S.K.Goutam. "The EB-Anubad translator: A hybrid scheme". *Journal of Zhejiang University Science*, Vol.6, 2005, pp.1047-1050.

[9] D.S Parikh P.Bhattacharyya "Interlingua Based English Hindi Machine Translation and Language Divergence", *Machine Translation* , Vol.16, 2001, pp.251-304.

[10] R.M.K. Sinha. "A hybridized EBMT system for Hindi to English Translation". *CSI Journal*, volume 37 no. 4, 2007, pp.3-9.

[11] 10. R.M.K. Sinha. "Designing Multi-lingual Machine- Translation System: Some Perspectives". International Conference on Machine Learning: Models, Technologies & Applications (MLMTA 2007), 2007 , pp. 244-249.

[12] 11. S..M Idicula, D. S. Peter. "A morphological processor for Malayalam language". *South Asia Research*. vol. 27 (2): 2007, pp.173-186.

[13] 12. L. Pandian, T.V.Geetha "Morpheme based Language Model for Tamil Part of Speech Tagging" *Polibits* (38) , 2008, pp.19-26 .

[14] 13. L.R.Nair, D.S. Peter. "Development of a rule based learning system for splitting compound words in Malayalam language". IEEE Recent advances in intelligent and computational systems(RAICS), 2011, pp.751-755.

[15] 14. L.R.Nair, D.S. Peter, “Shallow parser for Malayalam Language using finite state cascades”, 4th international congress on image and signal processing, China, 2011, pp.2464-2467.

[16] 15. S. Abney. “Partial parsing via finite state cascades”. *Journal of Natural Language Engineering*, **2**(4), 1996, pp. 337-344.

[17] 16. D.Jurafsky ,J.H Martin. *Speech and natural language processing.* India: Pearson Education, 2000,pp 657-671.

[18] 17. E. Rich, K. Knight, S. B Nair. *Artificial Intelligence*. New Delhi,India: The Tata McGraw Hill, 2009 pp 295- 300.

[19] 18. S.L.Devi, P.Pralayankar. “Verb Transfer in a Tamil to Hindi Machine Translation System”. International Conference on Asian Language Processing. Harbin, China, 2010, pp.261-264.

# Implementation of Urdu Probabilistic Parser

**Neelam Mukhtar** *sameen_gul@yahoo.com*
*Department of Computer Science*
*University of Peshawar, Pakistan*

**Mohammad Abid Khan** *abid_khan1961@yahoo.com*
*Department of Computer Science*
*University of Peshawar, Pakistan*

**Fatima Tuz Zuhra** *fateeshah@yahoo.com*
*Department of Computer Science*
*University of Peshawar, Pakistan*

**Nadia Chiragh** *nadiachiragh@yahoo.com*
*College of Home Economics*
*University of Peshawar, Pakistan*

**Abstract**

The implementation of Urdu probabilistic parser is the main contribution of this research work. In the beginning, a lot of Urdu text was collected from different sources. The sentences in the text were subsequently tagged. The tagged sentences were then parsed by a chart parser to formulate the rules. In the next step, probabilities were assigned to these rules to get a Probabilistic Context Free Grammar. For Urdu probabilistic parser, the idea of shift-reduce multi-path strategy is used. The developed software performs the syntactic analysis of a sentence, using a given set of probabilistic phrase structure rules. The parse with the highest probability is selected, as the most suitable one from a set of possible parses produced by this parser. The structure of each sentence is represented in the form of successive rules. This parser parses sentences with 74% accuracy.

**Keywords:** Urdu Probabilistic Parser, Urdu PCFG, Results of Urdu Probabilistic Parser.

## 1. INTRODUCTION

A lot of information about words and syntactic constructions are considered in parsing a human language [1]. Syntactic parsing has been widely studied with the help of different methods, including statistical parsing [2, 3, and 4] and linguistic-based methods [5, 6].

Statistical parsers are gaining popularity every day due to their noticeable accuracy and efficiency. A number of different statistical parsers are already developed by the natural language processing community [7, 8 and 9]. The main idea is to assign probabilities to the grammatical rules. “However, in practice, the probability of a parse tree being the correct parse of a sentence depends not just on the rules which are applied, but also on the words which appear at the leaves of the tree” [10].

Apart from Perso-Arabic script, the morphological system of Urdu is also making this language a highly challenging language because it has inherent grammatical forms and it has borrowed vocabulary from different languages such as Arabic, Persian, Turkish and the native languages of South Asia [11]. It is having a complex grammar with a complex script. The increasing use of Unicode characters and internationalization of software provides opportunities and ways to make research possible in this field [12].

In Urdu, research is going on from different point of views such as creating an Urdu corpus [13, 14] and tagging the Urdu corpus [15]. Researchers have proposed different tagsets for Urdu whose number of tags is ranging from 10 [16] to 350 [17].Now, one of the demanding areas is the parsing of an Urdu

corpus. Considerable amount of work has not yet been done in this direction. An efficient and accurate parser is therefore needed to parse Urdu corpus of natural text. A probabilistic parser is needed to parse Urdu sentences for adequate efficiency and accuracy compared to traditional rule-based parsers. Before developing such a parser, a Probabilistic Context Free Grammar (PCFG) is a pre-requisite.

Recently, a PCFG is developed for Urdu by taking Urdu tagged sentences from different sources [18]. After completing the pre-requisites of our objective, a new algorithm for Urdu probabilistic parser is created [19]. This algorithm is based on the idea of multi-path shift-reduce-strategy [20]. The algorithm is successfully implemented here thus resulting in an efficient Urdu probabilistic parser. This parser is tested by providing different Part-of-Speech tagged Urdu sentences as input. The parser successfully parses most of the sentences. The output from the parser is generated in the form of phrase based successive rules (resulting in a successful parse thus showing the structure of the sentence) with the highest probability. These rules clearly show the structure of the input sentence. Work in different sections of the research paper is organized as follows:

In section 2, the already developed PCFG is discussed briefly. Section 3 provides a view of the developed algorithm for Urdu Probabilistic parser and shows the implementation steps. Section 4 confirms the success of the parser by providing the results of the parser. Section 5 throws light on conclusion and future work.

## 2. URDU PROBABILISTIC CONTEXT FREE GRAMMAR

The sentences in the tagged text, available on the website of Center for Research in Urdu Language Processing (CRULP) under Urdu-Nepali-English Parallel Corpus project, are mostly long and ambiguous. These sentences are thus complex from parsing point of view. Therefore, apart from taking some complex sentences from the tagged corpus by CRULP (www.crulp.org), some additional data was also collected. Specifically, the focus was on Urdu aqwaal-e-zareen, mazameen and mini-kahanian written by famous authors such as Saadat Hassan Minto, Ibne Inshah and Pitras Bukhari.

Text was POS tagged by utilizing the annotator provided by CRULP. The tagset with 46 tags developed by CRULP as part of a project for developing Urdu-Nepali-English parallel corpus was used for text tagging. After acquiring the tagged data, rules were developed by utilizing the chart parser [21]. Context Free Grammar (CFG) for Urdu was thus developed.

A PCFG is a CFG where each production is assigned a probability. This probability is assigned to each rule by the ratio of the number of occurrences of a rule to the total number of occurrences of that particular phrase. Probabilities were assigned to the rules in the CFG to obtain Urdu PCFG [18].

Part of the table showing Urdu PCFG, with 127 rules is given below:

| **Rules** | **Probabilities** |
|---|---|
| S →NP VP | 0.9637 |
| S →PP VP | 0.0167 |
| S →PP NP | 0.0019 |
| S →VP NP | 0.0109 |

## 3. THE ALGORITHM AND ITS IMPLEMENTATION

The two most common types of parsing are top-down and bottom-up, though there are parsing algorithms that are of other types and there are some that are a combination of these two [22]. One such hybrid type is left-corner parsing. In left-corner parsing, top-down processing is combined with bottom-up processing to avoid going in a wrong way that may result (sometimes) when purely top-down or bottom-up technique is used.

A top-down parser starts its processing with the ‘start symbol’ S (sentence) and expands it by applying the productions until the desired string is reached. In bottom-up parsing, the sequence of symbols are

taken and compared to the right hand side of the rules. So the parser starts with the words (from the bottom) and attempts to build up a tree from the words until S is obtained [23].

Multi-path shift-reduce parsing (bottom-up parsing) keeps multiple transition paths during decoding [20]. It allows several best derived states after each expansion. The idea of multi-path shift-reduce parsing is used in developing algorithm for Urdu probabilistic parser. This idea is developed as part of this work. The above mentioned authors have only discussed theoretical concepts in their paper. There is no clue about the type of data structure(s) that is used in the implementation of the parser. In this research work, different data structures are used and the algorithm is implemented. Two different methodologies for rule storage are compared.

The special features of this work are:

a. A structured array is used for storing the results of multiple parses of the same string, where at each location there is:
    1. Identification whether to take shift action or reduce action.
    2. A separate input buffer.
    3. A separate processing stack.
    4. A separate output stack.
    5. A variable for calculating the total probability of the parse in the cell of the array.
b. The output goes to output stacks.

The algorithm for Urdu probabilistic parser [19], implemented here has four parts, i.e. a, b, c and d. Part a is the main algorithm whereas parts b, c and d are sub-algorithms. Algorithm a is the main algorithm for Urdu probabilistic parser. Algorithm b shows the procedure for checking/comparing the input on the top of the processing stack with the right hand side of the rules. Algorithm c is used for copying items onto a new processing stack. Algorithm d is used for finding the highest probability.

The algorithm discussed above is implemented using C# and XML. The parser produces a single representation, if the sentence is syntactically unambiguous. First, each configuration is represented as a rule and each rule is assigned a weight according to how often that configuration appears. These weighted rules are then used in parsing by calculating the total probability of the rules that are used in parsing. The parse with the highest probability is selected.

A user-friendly environment is provided for communication where a user inputs an Urdu file in the form of a collection of Part-of-Speech tagged sentences. The parser parses the text (by creating 200 stacks for each sentence) and produces the structure of each sentence in the form of successive rules as an output. In the beginning, rules were read from the text file but the system was unable to restrict the already used rules in previous stacks to be copied again in the new stacks within the same step. To solve this problem, rules are now read from the database table instead of a text file. Here identifiers are assigned to each rule so that the next stack will not use the already used rule again within the same step.

## 4. RESULTS

The structure of a sentence is shown by displaying the sequence of phrases that are used one after another in that particular sentence. A total of 100 sentences 22 long (having more than 10 words in a sentence) and 78 short are given as input to the parser. The parser parsed successfully 74 sentences. It failed to parse the remaining 26 sentences. These 26 sentences were kept separately and were carefully examined. Out of these 26 sentences, 15 sentences are long sentences. The remaining 11 sentences are short but ambiguous which is the reason for their unsuccessful parsing. The results obtained here cannot be compared with some other research in Urdu, because up to the knowledge of the authors, it is the first probabilistic parser developed for Urdu. The results from the parser, after processing the following POS tagged sentence, are shown in FIGURE 1:

```
<s>
 دوستوں<w POS= “NN”/>
 سے<w POS= “CM”/>
 محروم<w POS= “JJ”/>
 ہو<w POS= “VB”/>
 گیا<w POS= “AUXA”/>
 ہوں<w POS= “AUXT”/>
</s>
```

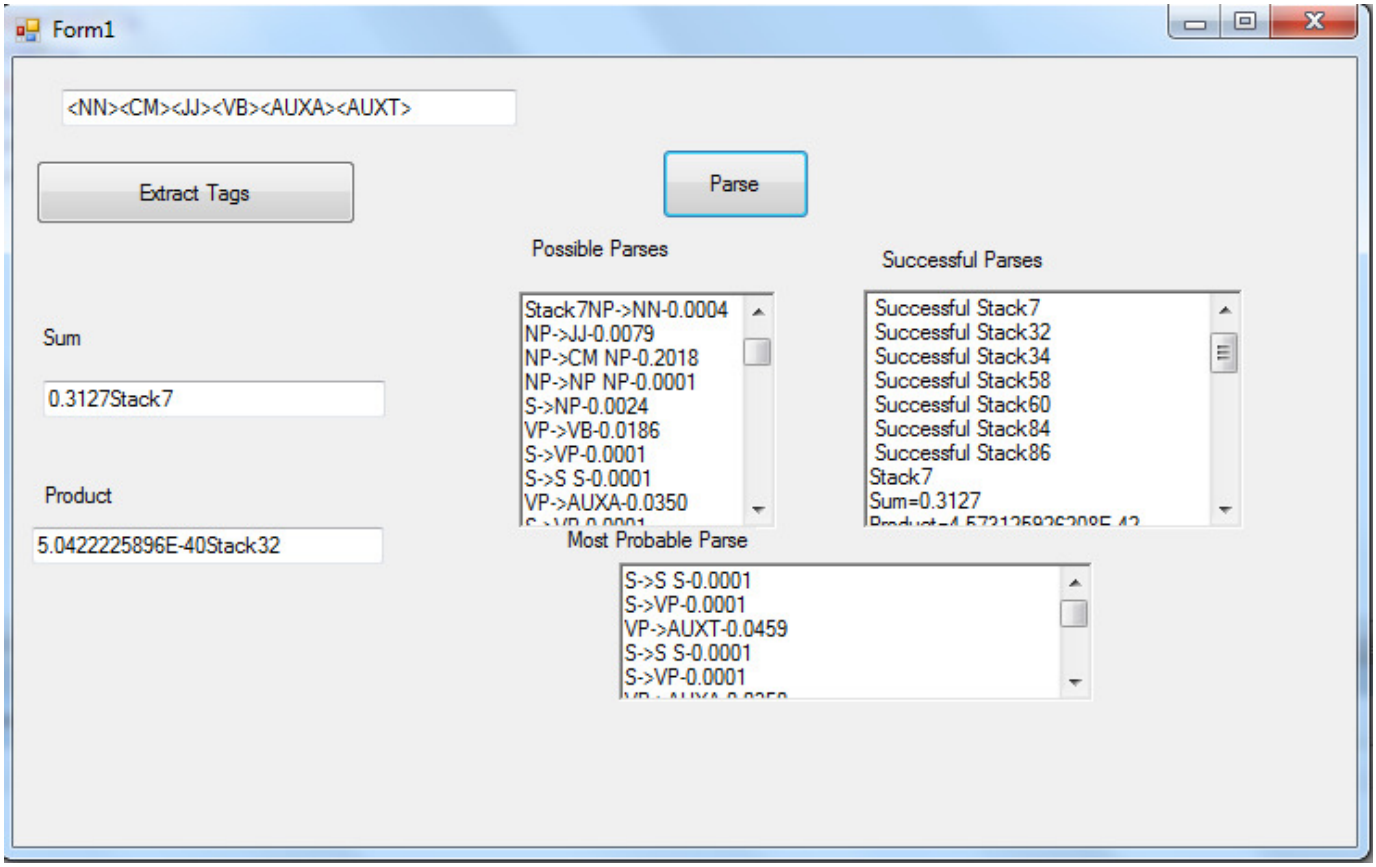



**FIGURE 1**: Complete output of a successfully parsed sentence.

Tags can be extracted from the sentence, when the button named “Extract Tags” is clicked (shown in FIGURE 1). When the button “Parse” is clicked, this parser provides structure of the sentence in the form of rules in four steps. Firstly, all possible parses of the processed sentence are shown under the heading “Possible Parses”. Here even unsuccessful parses are displayed. The stacks so far created and rules used are displayed. In case of a successful parse “Sc” is displayed at end of the rules in a stack showing that this particular stack is having a successful parse. Secondly, successful parses are shown under the heading “Successful Parses”. A complete list of successful parses with the stack number is provided here along with figures for the sum of probability and product of probability of the rules used. Finally, stack with the highest probability is considered as the correct parse of the sentence. The successive rules are displayed under the heading “Most Probable Parse”. One can easily draw a parse tree for the sentence from these successive rules. An example of one such sentence based, on the output by the Urdu probabilistic parser, using the POS tagged text below, is provided in FIGURE 2.

Successive rules provided by Urdu probabilistic parser.

S →NP-0.0024
NP →NNP NP-0.0688
NP →NNP NP-0.0688
NP →NNPC- 0.0051

<s>
<w POS= “NNP”>محمّد</w>
<w POS= “NNP”>اسد</w>
<w POS= “NNPC”>اللہ</w>
</s>

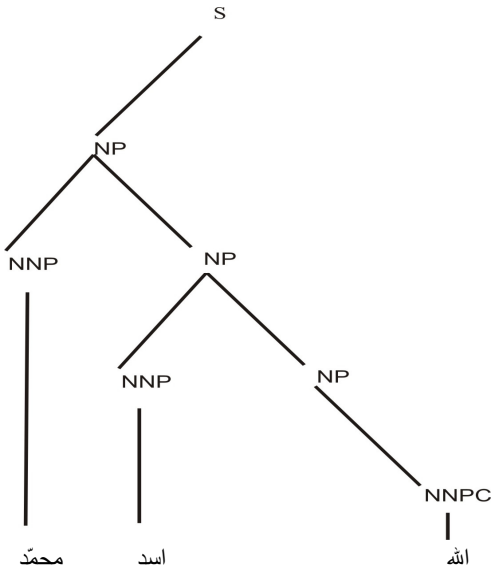



**FIGURE 2:** Parse tree for the POS tagged text.

In FIGURE 2, the parse tree is showing the structure of the sentence by utilizing the successive rules provided as output by Urdu probabilistic parser. If a sentence cannot be parsed successfully then “No successful parse” message is displayed in “Most Probable Parse” section as shown in FIGURE 3.

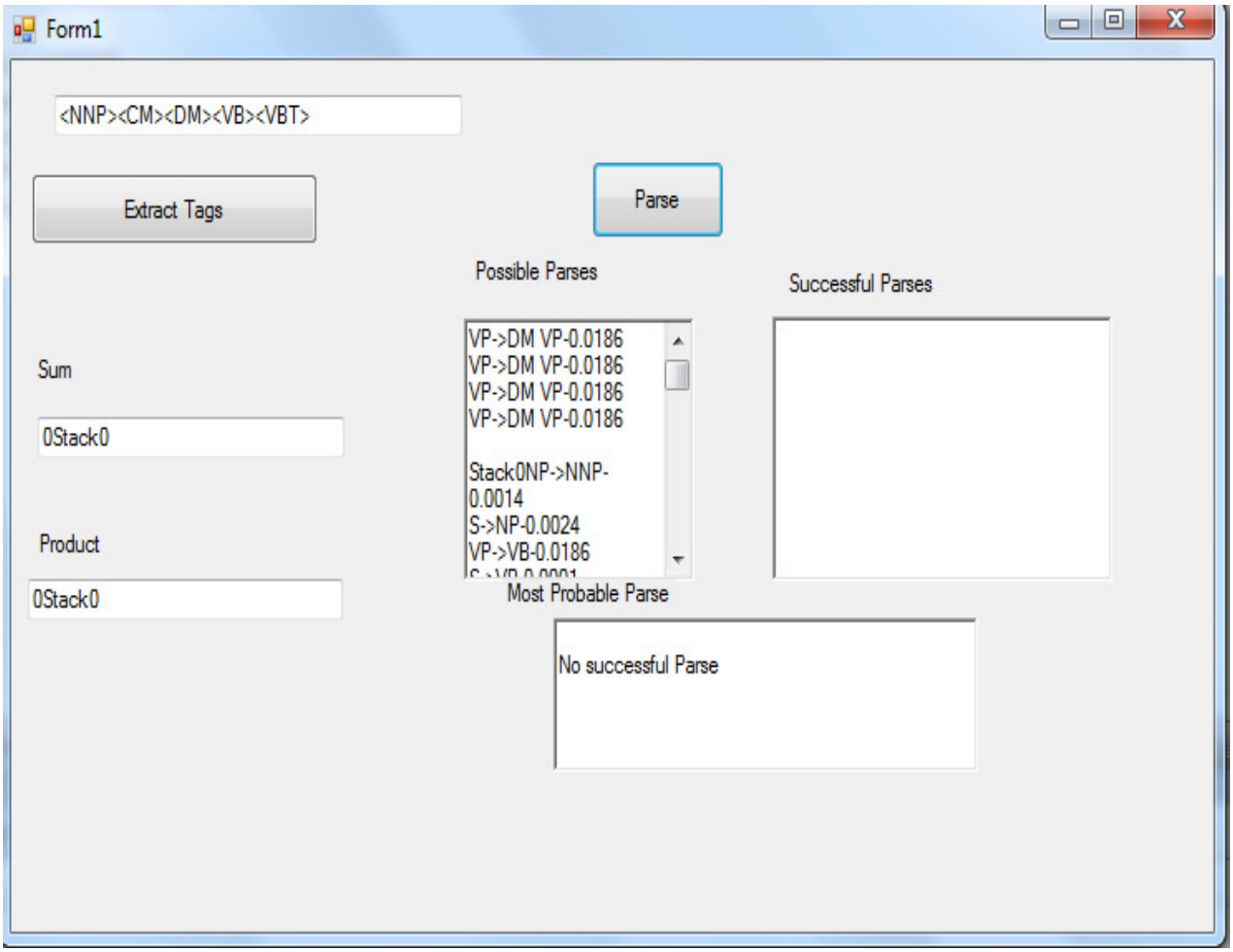



**FIGURE 3:** Sentence is not parsed successfully.

A section of the table showing the successfully parsed sentences by the Urdu probabilistic parser is provided in TABLE 1.

| Sentence | Successful parse |
|---|---|
| <s><br><w POS="NNP">محمّد</w><br><w POS="NNP">اسد</w><br><w POS="NNPC">اللہ</w><br></s> | NP →NNPC- 0.0051<br>NP →NNP NP-0.0688<br>NP →NNP NP-0.0688<br>S →NP-0.0024<br>Highest probability= 0.1451 |
| <s><br>دوستوں<w POS="NN"/><br>سے<w POS="CM"/><br>محروم<w POS=<br>"JJ"/><br>ہو<w POS="VB"/><br>گیا<w POS="AUXA"/><br>ہوں<w POS="AUXT"/><br></s> | NP →NN-0.0004<br>NP →JJ-0.0079<br>NP →CM NP-0.2018<br>NP →NP NP-0.0001<br>S →NP-0.0024<br>VP →VB-0.0186<br>S →VP-0.0001<br>S →S S-0.0001<br>VP →AUXA-0.0350<br>S →VP-0.0001<br>S →S S-0.0001<br>VP →AUXT-0.0459<br>S →VP-0.0001<br>S →S S-0.0001<br>Highest probability= 0.3127 |

**TABLE 1:** Output of the parser providing the rules

## 5. CONCLUSIONS AND FUTURE WORK

The parser mostly parses short sentences correctly. While parsing a long sentence (when the number of words exceeds 10), the parser usually fails. The parser takes a lot of time while parsing a highly ambiguous sentence. The parser consumes more memory by creating a number of stacks for highly ambiguous sentences. Sometimes, it fails to parse an ambiguous sentence. The reason is ambiguous grammar. By disambiguating the grammar the performance of the parser can be improved a lot.

Mostly this Urdu probabilistic parser can parse ambiguous sentences successfully. The chart parser, that was used in testing phase, usually failed to parse a sentence when a rule with left recursion (NP →NP NP) was required for parsing. This Urdu probabilistic parser can successfully parse a sentence even by using the rules with left recursion.

While considering the 100 tested sentences by Urdu probabilistic parser, the success rate of the parser is 74%. Usually short sentences (i.e. less than 10 words in a sentence) are parsed successfully. It may be concluded that to a large extent the success of this parser is dependent on the length of the sentence. The shorter the sentence, the higher is the chance of being successfully parsed by the parser.

For further improvement, the number of stacks is increased from 200 to 500. The performance of the parser in both the cases is compared. It is observed that increase in the number of stacks has a very small effect on the accuracy (almost negligible) of the parser. The speed of the software is decreased a lot as the software has to create 500 stacks now for each sentence. Considering speed as an important factor, parser with 200 stacks is more suitable.

The use of an Urdu tagger (so that it will tag the sentences automatically at the moment they are entered) will increase the efficiency of the parser. The use of a tree generator that will create a parse tree from the rules will make the software more efficient. The use of different data structures such as linked lists, instead of stacks and arrays will decrease the memory requirement of the algorithm.

## REFERENCES


[1] B. Sagot and E. de la Clergerie. "Error Mining in Parsing Results". Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, Sydney, 2006, pp. 329–336.

[2] E. Charniak. "Statistical Parsing with a Context-Free Grammar and Word Statistics". Proceedings of the 14th National Conference on Artificial Intelligence, MIT Press, 1997.

[3] M. J. Collins. "A New Statistical Parser Based on Bigram Lexical Dependencies". Proceedings of ACL 96, 1996.

[4] D. M. Magerman. "Statistical Decision- Tree Model for Parsing". Proceedings of the 33rd Annual Meeting of the ACL, 1995.

[5] S. Abney. "Partial Parsing via Finite-State Cascades". John C. Ed. Workshop.Robust Parsing (ESSLLI'96), 1996, pp. 08-15.

[6] S. A'it-Mokhtar and J-P. Chanod. "Incremental Finite-State Parsing". Proceedings of the 5th Conference on Applied Natural Language Processing, 1997.

[7] M .J. Collins. "Head-driven statistical models for natural language parsing". Ph.D. thesis, 1999.

[8] E. Charniak."A maximum-entropy inspired parse", Proceedings of the First Meeting of The North American Chapter of the Association for Computational Linguistics, Seattle, WA, 2000, pp. 132–139.

[9] S. Petrov, L. Barrett, R. Thibaux. and D. Klein. "Learning accurate, compact and interpretable tree annotation". Proceedings of ACL, 2006.

[10] C. Lakeland and A. Knott. "Implementing a Lexicalized Statistical Parser". Proceedings of the Australasian Language Technology Workshop, Macquarie University, Sydney, 2004.

[11] M. Humayoun, H. Hammarström and Ranta. "Urdu Morphology, Orthography and Lexicon Extraction". CAASL-2, the Second Workshop on Computational Approaches to Arabic Script-based Languages, LSA 2007, Linguistic Institute, Stanford University, 2007.

[12] M. Humayoun. "Urdu Morphology, Orthography and Lexicon Extraction". Master thesis, Department of Computer Science and Engineering, Chalmers University of Technology and Goteborg University, 2006.

[13] H. Samin, S. Nisar and S. Sehrai. Project: "Corpus Development". BIT thesis, Department of Computer Science, University of Peshawar, Peshawar, Pakistan, 2006.

[14] D. Becker and K. Riaz. "A Study in Urdu Corpus Construction". Proceedings of the 3rd Workshop on Asian language resources and international standardization, 2002.

[15] W. Anwar, X. Wang, Luli and Wang. "Hidden Markov Model Based Part of Speech Tagger for Urdu". Information Technology, 2007, *Vol.6*, pp. 1190-1198.

[16] R. L. Schmidt. "Urdu: an essential grammar". Rout-ledge, London, UK, 1999.

[17] A. Hardie. “The computational analysis of morph syntactic categories in Urdu”. Ph.D thesis, Lancaster University, 2003a.

[18] N. Mukhtar, M. A. Khan and F. Zuhra. “Probabilistic Context Free Grammar for Urdu”, Linguistic and Literature Review (LLR), 2011, 1(1).

[19] N. Mukhtar, M. A. Khan and F. Zuhra. “Algorithm for developing Urdu Probabilistic Parser”, International journal of Electrical and Computer Sciences IJECS-IJENS 12(3), pp. 57-66, 2012.

[20] W. Jiang, H. Xiong and Q. Liu. "Multi-Path Shift-Reduce Parsing with Online Training".CIPS ParsEval, Beijing, November, 2009.

[21] F. Zuhra. "Pashto Chart parser". Unpublished paper, Department of Computer Science, University of Peshawar, Pakistan, 2010.

[22] G. Sandstrom. Survey paper, "Parsing and Parallelization", 2004.

[23] B. M. Bataineh and E. A. Bataineh. "An Efficient Recursive Transition Network Parser or Arabic Language".Proceedings of the World Congress on Engineering, London, U.K, 2009.

# An Approach for Knowledge Extraction Using Ontology Construction and Machine Learning Techniques

**Dhanasekaran K** *dhana_mec5684@yahoo.co.in*
*Research scholar/Computer Science and Engineering*
*Info Institute of Engineering*
*Anna University of Technology*
*Coimbatore-641 107, India*

**Rajeswari R** *rreee@gct.ac.in*
*Assistant professor/Department of Electronics and Instrumentation Engineering*
*Government College of Technology*
*Anna University of Technology*
*Coimbatore, India*

**Abstract**

In recent research, Ontology construction plays a major role for transforming raw texts into useful knowledge. The proposed method supports efficient retrieval with the help of ontology and applies combined techniques to train the data before taking into testing process. The proposed approach used the phrase-pairs to extract useful knowledge and utilized data mining techniques and neural network approach to express the knowledge well and also it improves the search speed and accuracy of information retrieval. This method avoids noise generation by analyzing the relevancy of tags to the retrieval process and shows somewhat better recall value compared to other methods. In this approach an optimized reasoner applied to reduce complexity in the key inference problem. The formulated ontology can help clearly expressing its meaning for various concepts and relations. Due to the increasing size of ontology repository, the matching process may take more time. To avoid this, this method forms a hierarchical structure with semantic interpretation of data. The system designed to eliminate domain-dependency with the help of dynamic labeling scheme using ontology as a base. In this paper, our proposed models were presented with ontology description using Ontology Web Language (OWL).

**Keywords:** Back Propagation, Domain Ontology, Knowledge Extraction, Agricultural Environment.

## 1. INTRODUCTION

Agricultural sector plays an important role in increasing economy of any nation. With this in mind, researchers nowadays, focusing on solving problems that causes either damages or disease with respect to plant production, rice production, etc., and they are trying to solve problem in respect of the issue for improving the productivity and increasing an economy and reducing the cost that is to be invested in production [1].

This method allows us to identify various qualifiers for retrieving information which are relevant to the user query. In survey there is a evidence that there are no efficient construction of agriculture ontology method that is available to address various issues which degrades growth in multiple dimension [1][2].In the proposed method, optimal tagging will be performed based on the weights assigned to the items in the list during analysis stage. Initially the data is trained with the help of back propagation method and learning is performed on those data to produce valuable data for the testing process. Finally, the data which are stored at various nodes at various levels in the hierarchy in some order will be processed and retrieved with improved recall value.
The mapping is carried out by checking for correspondence between input vectors and the expected output vectors. In addition, the similarity and differences in the concepts and relations

are identified to eliminate redundancies and ambiguity that are present in the knowledge repository.

In this method, various guidelines and rules are formulated by associating relevant features which are extracted from the hierarchical structure. This will support presenting some advisable ideas to the farmer and suggests some methods for preventing damages or controlling disease in cultivation of varieties, planting, etc., the new method improve efficiency in information extraction process with the help of a generated ontology which adopts some rules and criteria recommended by the researchers [4].

Ontology is mainly constructed as a formal specification of various concepts and relations between properties of those concepts. Ontology is defined as partial specification of conceptual vocabulary used for formulating knowledge-level theories about a domain of discourse. Ontology is applied in domains like natural disaster management system, medicine, military intelligence, cooking, enterprise, jobs, agriculture, Wikipedia, automobiles and so on. Ontology is also expressed as a formal representation of knowledge by a set of concepts within a domain and the relationship between these concepts. Ontology consists of four main components to represent a domain. They are:
i. Concept represents a set of entities within a domain.
ii. Relation specifies the interaction among concepts
iii. Instance indicates the concrete example of concepts within the domain
iv. Axioms denote a statement that is always true.

Let us take an example of a wine ontology and look at its components. The concepts of the wine ontology are, “Winery, Wine, Wine descriptor, Wine color, etc”. The relationships are given as Winery *produces wine*, wine *has* wine descriptor. The instances of wine color can be red, rose and white. The axiom in this example is ‘a winery must produce at least one type of wine’.
Ontologies can be constructed using three different approaches; single ontology approach, multiple ontology approach and hybrid ontology approach [22].

The single ontology approach is the simplest of all and it uses single global ontology for all information sources, which shares the vocabulary and the terminology to specify the semantics. The limitation of this approach is that it does not provide a perfect solution for information integration. This limitation has paved way for multiple ontology approach, where each information source is described by its own ontology thus each source will have its own local ontology. The main drawback of this approach is the construction of individual ontology. The hybrid ontology approach is the combination of single and multiple approaches [22].

In general, ontology construction could be done in three ways [23]:

- Manual: Ontology is constructed manually.
- Semi automatic: Human intervention is needed during ontology process.
- Fully automatic: The system takes care of the complete construction.

Ontology construction involves six basic steps.

1. Ontology Scope
2. Ontology capture
3. Ontology encoding
4. Ontology integration
5. Ontology evaluation
6. Ontology documentation.

It gives semantic expression for each and every term in the conceptual framework by means of combining relevant items together with the help of a Natural Language Processing tool. In my

study part, I could see that some methods are developed to construct ontology that refers to business model and data model for petroleum exploration and production domain feature, etc., [2][5][6].In ontology research, no one focused on the related research of agricultural domain that analyzes how planting method, use of ingredients, irrigation method affects land quality, productivity, as a result economy.

The Extractor in this method will retrieve the factor that characterizes planting, cultivation of varieties, various methods in irrigation, etc., It uses various concepts and relations for the core vocabularies of the considered field. With these, it makes it possible to regulate the extension of features and an ontology is used to describe them using XML as a base and OWL ontology language as a description language.

## 2. LITERATURE REVIEW

In the year 2009,Amal Zouaq and Roger NKambou have published a paper on "Evaluating the generation of domain Ontologies in the Knowledge Puzzle Porject".The author described the procedure to extract concept maps from texts that are followed by TEXCOMON,Knowledge Puzzle Ontology Learning tool. In this paper, they are evaluated ontology in three dimensions: structural, semantic and comparative. In structural evaluations, ontology is considered as graph based on a set of metrics. Semantic evaluation is carried out using human expert judgement.Finally comparative evaluation is done by comparing the output of current tools and new tools. This task has used the same set of documents for all cases.

They compared the ontological output in terms of concepts, attributes, hierarchical and non-taxonomic relationships. The method produced more interesting concepts and relationships but failed to avoid a lot of noise generation by lexico-syntactic patterns and their methods. They suggested developing method for improving the patterns.

Moreover, the OWL Java API of their project improved in terms of processing time. This paper consists of a future direction towards automating ontology evaluations in order to solve a number of problems such as., Ontology-learning, population, mediation and matching.

In the year 2004, Marta Sabou has published a paper on "Extracting Ontologies from Software Documentation: A Semi-automatic method and its evaluation". In his approach, he used software APIs to build domain ontology by extracting types of method functionalities. In that method, a small corpus is used for applying statistical techniques. The author has described that there is a need to enhance the corpus and to develop a better extraction method that suits the small corpora. This method is encouraging towards building an Ontology extraction method from software APIs.

In the year 2010, B.Saleena, Dr.S.K. Srivatsa has published a paper on "A Novel Approach to develop a self-organized Domain Specific Search mechanism for Knowledge Acquisition using Ontology". The authors have created a search method for semantic web in order to design a self-organized system to retrieve information about a particular topic based on user interest in learning.

For this they created a knowledge library for DBMS domain using Ontology and Knowledge management technologies. And then they followed a strategy to group the relevant information for the user in a single search. The search is implemented based on keyword which is a time-consuming process. The system has been developed in JAVA 2 API with OWL API for semantic web. It retrieves the inter-related contents, prerequisites and further readings needed to understand the topic as per user's interest. With the help of this, an e-learning framework is developed using Ontology based knowledge retrieval.

This paper included future work to enhance the system for various domains and to add a large set of functionalities to the UI screen to improve the user-friendliness. Also it has been suggested to develop a method for automatic extraction of information.
In the year 2009, Song Jun-feng, Zhang Wei-ming, Xiao Wei-dong, Xu Zhen-ning has published their work "Study on Construction and Integration of Military Domain Ontology, Situation Ontology and Military Rule Ontology for Network Centric Warfare". The author have discussed that there is a need for knowledge infrastructure in network centric warfare in order to transform information into knowledge.

In this paper, they proposed approach to construct all three kinds of ontology mentioned above. They also addressed the integration approach using all these approaches. Then they have constructed scenario based knowledge infrastructure fragment using proposed approaches and techniques. They said that current research works are few in this respect, in future study; they are going to study what kinds of other component ontology are needed for the knowledge infrastructure. They also planned to implement experimentation and at present, they would like to use protégé basic tool.

The conceptual graph of ontology language (OML) lacks precise semantics. OWL is a new synthesis of research on ontology language. The expressiveness of all the languages are very limited and key inference problem has most complexity. So, there is a need for optimized reasoners.

In the same year, Zhang Rui-ling, XU Hong-Sheng have published a paper on "Using Bayesian Network and Neural Network Constructing Domain Ontology". In this paper, they have addressed that the current ontology construction methods have limitations. They are: 1) Requirement for human labor 2) Domain restrictions. To avoid these problems, they developed an approach to construct ontology based on a novel method which contains Projective Adaptive Resonance Theory (PART) neural network and Bayesian Network Probability theorem.

Their system could acquire key terms automatically. Finally it reasons out the complete terms in the classification framework in order to construct domain ontology. The ontology is stored using a Resource Description Framework (RDF).The Semantic Web can be deployed based on the rapid and efficient construction of the ontology. Some of the features of this work are: the PART architecture is included to overcome the lack of flexibility in clustering, and in the web page analysis, WordNet deals with the lack of knowledge acquisition.

Finally they said that there is a need to improve the precision of term location. Due to the accumulation of the number of documents in the ontology repository, the similarity calculation takes more time. This is unavoidable. So, if we build an approach to form a hierarchy of clusters, it will solve the problem. The current methods can build only a partially automated classification of terms". This involves a time-consuming process and costly procedure.

In the same year, Yi Zhang, Li Tan, Jie Liu, ChangChang Yu has published a paper on "A Domain Ontology Construction method towards Healthy Housing". Due to the presence of various domains, there is no efficient framework for ontology construction. The authors have introduced an improved ontology construction method. In this method, they used graphic language to represent domain knowledge for research domain. The system evaluated and verified the correction of relationships and hierarchy for constructed ontology. The proposed method for Healthy Housing solves the problems such as lack of semantic expression and understanding of expert knowledge.

In future work, it is stated that the research can be done to utilize this Housing ontology in the Healthy Housing Intelligent Synthesized Evaluation System by mapping and matching domain ontology. Then it will be possible to realize the interchange between natural language of professional field and conceptual-level ontology language which can be understood by machines.

In the year 2010, Zhang Dan, hang Li, Jiang Hao have published their paper "Research on Semi-automatic Domain Ontology Construction". They have applied Data Mining method and word partitioning technique to construct semi-automatic domain ontology. At certain level, they could recognize the effectiveness and quality of the ontology. In this paper, they said that the semi-automatic approach still poses a problem because of the difficulties in constructing a common tool. The reasons are: choosing data source is manual, extracting compound words without considering the characteristics of language, and analyzing the grammatical components of sentence to conclude the relations among concepts. The authors said that the methods can be tried to address these issue.

## 3. ONTOLOGY CONSTRUCTION METHOD

Existing ontology methods are focused on acquiring concepts, properties and relations, and also focus on emphasizing the description of knowledge ontology. Although it can eliminate ambiguity based on user's judgment, it fails to apply analysis and expression evaluation for predicting accuracy of constructing ontology. This paper gives solution for this problem.

Steps in the process of Ontology construction are as follows:

**Step1: Domain Analysis Phase**
In this phase, determination of extensibility and reusability of domain ontology are taken place.

**Step2: Ontology Analysis and Design Phase**
In this phase, Agricultural domain ontology will be established by defining the hierarchy of concepts and relations between different activities and then translates the professional knowledge and raw data of the domain to commonly used information. This phase acquires semantic information about concepts, relations, actions, etc.

**Step3: Ontology Representation**
In this phase, Individual classes are created based on properties of the class and domain ontology is represented using XML and OWL ontology description language.

**Step4: Ontology Evaluation**
Here, the rule set is formed and inference mechanism is applied and then the accuracy and correspondence between input and output vectors are evaluated. Hence, it checks for the consistency of the constructed model. These steps are shown in Fig.1.

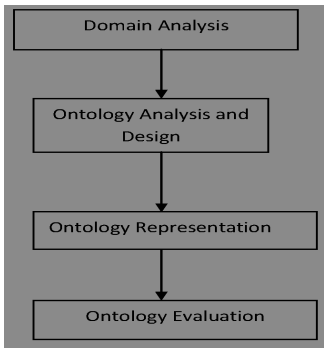



**FIGURE 1** The generic process steps of Agricultural ontology construction

The proposed approach shown in Fig.2 include ten simple criteria: lawfulness (i.e. frequency of semantic errors), richness (how many of the semantic features available in the ontology are used by the processing element and data mining algorithm), interpretability (do the phrase pairs used in the ontology also appear in Phrase Net?), consistency (how many concepts in the ontology are involved in inconsistencies),clarity (do the phrases used in the ontology have many senses in Phrase Net?), comprehensiveness (number of concepts in the ontology, relative to the average

for the entire library of ontologies), accuracy (percentage of false statements in the ontology), relevance (number of statements that involve semantic features marked as useful or acceptable to the user/agent), authority (how many other ontologies use concepts from this ontology), history (how many accesses to this ontology have been made, relative to other ontologies in the library/repository)[24].

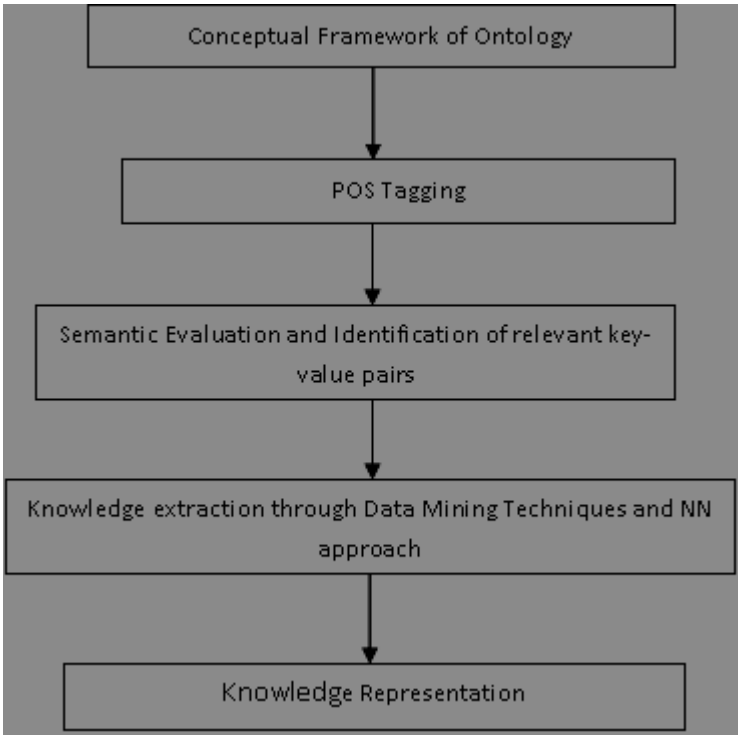



**FIGURE 2** Proposed model of Knowledge Extraction

## 4. CONSTRUCTION OF AGRICULTURE ENVIRONMENT CONTROL AND MANAGEMENT ONTOLOGY

### 4.1 Domain Analysis

In this phase, grouping of the expert's opinion and the definition of the domain consists of four higher-level concepts such as., agricultural environment standard, climate analysis standard, watering service or irrigation standard, and land use environment standard.

At the top-level more general concepts which describe the common properties are defined. On the other hand, at the lower level, more specific properties of the standards are defined which can be used as subclass. Its instance in the domain ontology plays a major role for sending input data and receiving output signal in order to perform snalysis, processing or other kinds of tasks.

### 4.2 Ontology Analysis and Design

The subclasses of core concepts are again divided into many lower level concept which supports extraction of correct knowledge for the users by comparing useful patterns of the original information. The entire structure of the environment is shown in Fig.2.

This method gives importance for generalizing the concepts into more specific concepts. Thus the constructed ontology will not have redundancies and ambiguity by finding the right classes, subclasses, instances, and properties. It supports getting better recall value for the user query.

In Fig.3, the subclass environment with appropriate relationships is derived among various conceptual elements. The agricultural environment standard has relationship with concepts such as., climate analysis standard, this in turn, allows us to express the concepts such as., temperature analysis, precipitation control, monitoring service, information management.

### 4.3 Ontology Representations

After the analysis and design of ontology, it will be represented using 5-tuple.They are defined as follows:
Definition: DOR :=( SC, RC, DFE, LLC, PLLC)
Where,
SC: Super Class which acts as root of the hierarchy
RC: Relationships among Concepts
DFE: Data and Functional Elements
LLC: Lower Level Concepts which represents some knowledge at the detailed level.
PLLC: Properties of Lower Level Concepts which represents some derived and specific properties of its own.

**1) Super Class (SC)**

This represents concepts of ontology by showing some top-level concepts like watering service setting, land use analysis and so on. Also it shows some attributes like planting date, location, yield and so forth.

**2) Relationships among Concepts (RC)**

The basic relationships in Agricultural domain ontology are shown in table1.

| Relation | Definition of relation |
|---|---|
| Kind-of | The inheritance of concepts |
| Part-of | The relation between mass and parts |
| Instance-of | The relation between instance and concept |

**TABLE** 1: The basic relations of Agricultural Domain

For example, in Fig.3, Temperature and Precipitation have Part-Of relationship with Weather class.

**3) LLC (Lower Level Concepts)**

The lower level mostly has kind-of character with its higher-level. In Fig.3, the Weather control and Management environment has been defined with properties and relationships as follows:

```
Def Category Weather control and Management
{
Property: Weather planning
      : Type string
      : Comment “plan of weather control”
Property: Management planning
      : Type string
      : Comment “plan of information management”
}
```

**4) PLLC (Properties of Lower Level Concepts)**

This represents implementation of class by writing class with derived values and some additional values of its own.

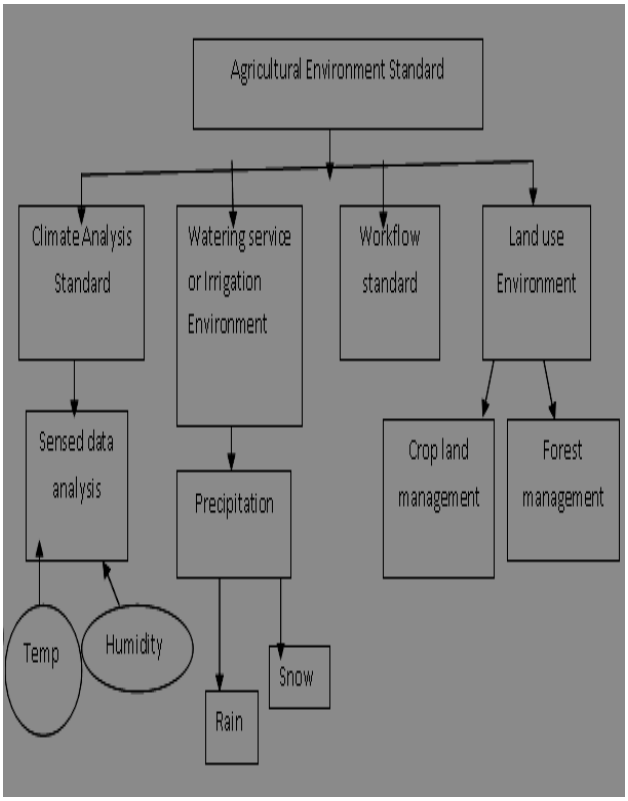



**FIGURE 2:** The structure of Agricultural environment standard

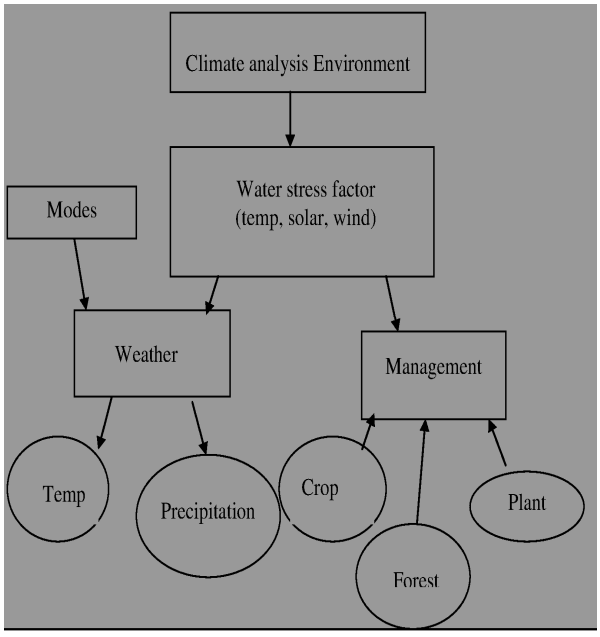



**FIGURE 3:** Relationship between weather and management

In Fig.4, considering Environmental factor, they can be classified as Nonbiotic factor determination and outcome planning.

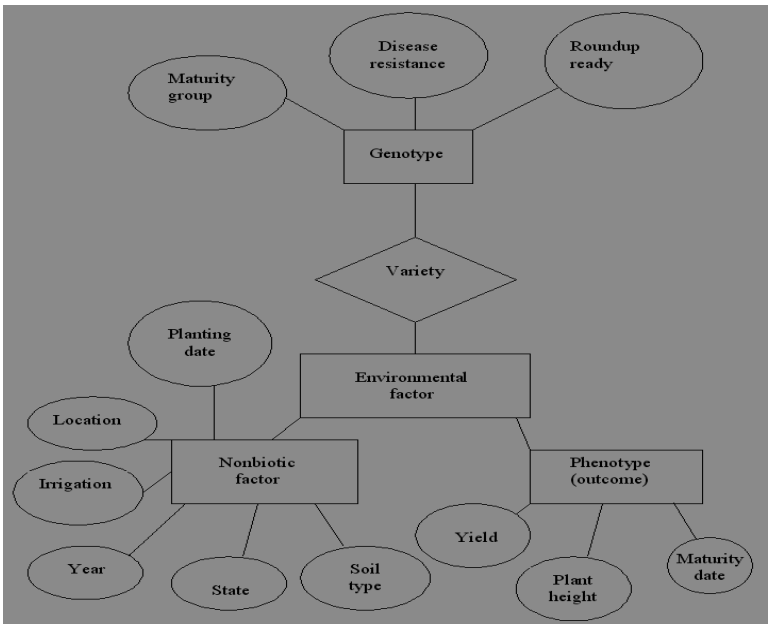



**FIGURE 4:** A model of agriculture agronomy

These factors may affect the environment and they are defined as follows:

Def Environment Analysis

{

Nonbiotic factor determination: Make decision for the parameter such as., planting date, location, irrigation, year, state, soil type, etc.,

Outcome generation plan: Make decision for generation of yield, maturity date, plant height.

}

In this paper, Protege tool used for establishing ontology by defining various attributes, activities, objects, and relationships. The land use environment is described as a subclass of Agricultural Environment Standard using OWL language as follows:

<? xml version=1.0"?>
<rdf: RDF
Xmlns:sqwrl=http://sqwrl.standard.edu/ontologies/built-ins/3.4/sqwrl.owl#
xmlns="file:/D:/MyFolder%18files/Protege3.4.1/Agricultural environment standard.owl#"
xml:base:"file:/D:/MyFolder%18files/Protege3.4.1/ Agricultural environment standard.owl">
<owl:Ontology rdf:about="file:/D:/MyFolder%18files/Protege3.4.1/ Agricultural environment standard.owl">
<owl:importsrdf:resource=http://sqwrl:standard.edu/ontologies/built-ins/3.4/sqwrl.owl/>
<owl:importsrdf:resource=http://sqwrl:standard.edu/ontologies/built-ins/3.3swrla.owl/>
</owl: Ontology>
<owl: Class rdf: ID="Sensed data analysis">
<rdfs: subClassOf>
<owl: Class rdf: ID="Climate analysis standard"/>
</rdfs: subClassOf>
</owl: Class>

**5) Ontology Testing**

The accurateness of constructed ontology is verified by comparing the MADRE method. The ontology is evaluated in two ways: 1) verification of relations 2) verification of hierarchy. Both of these verification compare the result that are obtained by inference mechanism of domain ontology in this paper and then if it satisfies the requirements then it will be added into the domain ontology library,otherwise,going back to the analysis phase, it finds new ontology and definition.

## 5. CONCLUSION

Ontology helps us to make knowledge acquisition and retrieval process in very easier manner. The existing methods and patterns for information retrieval do not allow us to accurately retrieve information so that various methods have been developed so far, to address issues in different domains. In this paper, we have discussed a new approach to produce somewhat better result for the selected problem in agricultural domain with the help of an efficient ontology construction, data mining and neural network approach. The method is developed to address disaster and disease control issues in the selected domain irrespective of tasks such as., planting, cultivation of varieties, irrigation, etc.It shows somewhat better performance compared to other methods and act as a new framework for agricultural domain. Also this work enables the user to utilize an ontology query method by using pair-wise tagging. The optimal pair-phrase is generated using Natural Language Processing tool focusing on identified key issues. The formulated method can clearly express its meaning for various concepts and relations. The limitations in the keyword or term based extraction is eliminated by implementing query transformation technique which generates intra-query value pair. This gives suggestions and lists previous histories for the user query by reducing the computation time and cost and producing the improved recall value. Although this method produces encouraging results with an agricultural data set, future work will involve libraries for testing the performance of the proposed method on other data set. It should also be possible to generalize our approach to learning simulataneously several subtrees in the ontology tree.

## 6. REFERENCES

[1] A.Thunkijjanukij, A.Kawtrakul, S.Panichsakpatana , U.Veesommai.” Lesson learned for ontology construction with Thai rice case study”. “in press”.*World Conference on agricultural information and IT*,2008,pp.495-502.

[2] W.Bi-long , H.Li.”Method of building petroleum exploration and production domain ontology”.“in press” *Computer Engineering and Applications*, 2009.pp.1-3.

[3] Z.Rui-ling, X.Hong-sheng.”Using Bayesian network and neural network constructing domain ontology”.“in press” *World Congress on Computer Science and Information Engineering*, 2009, IEEE 2008, pp.116-231.

[4] Guarino.N. “Formal ontology in information systems”.“in press”, in *Proc.IOS*,1998.

[5] Elena P,Sapozhnikova.” Multi-label classification with art neural networks”. “in press” *Second International Workshop on Knowledge Discovery and Data Mining*, 2009 IEEE, pp.144-147.

[6] X.LI, l.ZHAO, l.WU.”A feature extraction method using base phrase and keyword in Chinese text”. “in press” in *Proc. 3rd International Conference on Intelligent System and Knowledge Engineering*,pp.680-685, 2008 IEEE.

[7] S.Zhao R.Grishman,”Extracting Relations with Integrated Information Using Kernel Methods”, “in press” *in Proc.43rd Annual Meeting of the ACL*, pp. 419–426, Ann Arbor, Jun. 2005.

[8] A.Nag!, S. Biswas*, D. Sarkar*, P.P. Sarkar*, B. Gupta**.(2010,Jun.). “A Simple Feature Extraction Technique of a Pattern By Hopfield Network”, “in press” *International Journal of Advancements in Technology*, ISSN 0976-4860, Vol 1, No 1,pp.45-49.

[9] H.Liu1, 3, A.Abraham2, and B.Yue3," Nature Inspired Multi-Swarm Heuristics for Multi-Knowledge Extraction", "in press" *Advances in Machine Learning II*, pp. 445–466, 2010.

[10] Spyns,P."EvaLexon:Assessing triples mined from texts".*Technical Report 09*,STAR Lab,Brussels,Belgium,2005.

[11] Velardi,P.,et al."Evaluation of OntoLearn,a methodology for automatic learning of domain ontologies.In:Ont.Learning from Text:Methods Evaluation and Applications".*IOS Press*,2005.

[12] Weng,S.,Tsai,H,Liu,S.,and Hsu,C."Ontology Construction for information classification ,*Expert Systems with Applications*",pp.1-12.

[13] Christine W. Chan.(2004)"From Knowledge Modeling to Ontology Construction". *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*.

[14] G.Cui, Q.Lu, W.Li, Y.Chen.Corpus Exploitation from Wikipedia for Ontology Construction: pp.2125-2132.

[15] C.Chou, F.Zahedi, H.Zhoa."Ontology for developing websites for natural disaster management: methodology and implementation", 2008.

[16] Harith Alani."Position paper: Ontology construction from online ontologies".in *Proc. the 15th international conference on World Wide Web* . 2006,pp. 491 - 495.

[17] Antonio M. Rinaldi."An Ontology-Driven Approach for Semantic Information Retrieval on the Web". *In ACM Transactions on Internet Technologies*, 2009,Vol. 9, Article 10.

[18] X.Binfeng,Luo, Xiaogang P.Cenglin, H.Qian."Based on ontology: construction and application of medical knowledge base".*IEEE International conference on complex medical engineering*, 2007,pp.586- 589.

[19] A.Rafea, Hesham A. Hassan, M.Yehia Dahab."TextOntoEx: Automatic Ontology Construction from Natural English Text". *International conference of Artificial Intelligence and Machine Learning,* 2006.

[20] A.A. Barforoush, S.Soltani.( 2009)."Web Pages Classification Using Domain Ontology and Clustering". *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*,.

[21] A.Moreno and D.Sánchez.(2008)."Learning non-taxonomic relationships from web documents for domain ontology construction".*Intl.jl.of Data and Knowledge Engineering*.

[22] S. Hubner, H.Neumann, H.Stuckenschmidt, G.Schuster, T.Vogele, U.Visser, H.Wache."Ontology-Based Integration of Information-A survey of Existing Approaches".2001.

[23] M.A. Ismail, M.Yaacob & S.A.Kareem."Ontology Construction: An Overview".Malaysia, 2006.

[24] Thai National AGRIS Centre,Dept.of Soil Science,Faculty of Agriculture,Kasetsart University,Dept.of Computer Engineering,Kasetsart University,(2009)."Rice Production Knowledge Management:Criteria for Ontology Development".*Thai Journal of Agricultural Science*, Thailand.

# Language as a renewable resource: Import, dissipation, and absorption of innovations*

**Bengt-Arne Wickström** *wickstr@hu-berlin.de*
*Institut für Finanzwissenschaft*
*Humboldt-Universität zu Berlin*
*DE-10099 Berlin, Germany*
and
*School of Economics and Management*
*Free University of Bozen-Bolzano*
*IT-39100 Bozen-Bolzano, Italy*

**Abstract**

The structural stability of different languages subject to the import of external elements is analyzed. We focus on the temporal side of the different processes interacting to produce a change in the structure of the language. That is, the rate of import and dissipation of new elements is seen in relation to the rate at which a language absorbs such new elements into its structure. The analysis leads to a model that in the steady state is formally similar to the standard model used to analyze the extraction of renewable natural resources.

This model is applied to different sociolinguistic situations and we speculate about how the structural type of a language might influence its rate of adaptation of the external innovations and how the cultural and social status of the idiom (partially) determines the rate of import of such innovations. Conditions that might lead to attrition and decay of the linguistic system, are characterized and some policy implications are drawn.

The model is presented as a theoretical model with only a few illustrative simulations. However, the structure is such that it can easily be adapted to computational methods and used in simulations. With obvious extensions, sociolinguistically more complex (and realistic) situations can be modeled.

**Keywords:** Language contact, borrowing, language structure, language status, language shift, language attrition, language planning, renewable resource.

## 1. INTRODUCTION

Languages have always changed and influenced one another. The vocabulary of high-status languages, especially, has entered and enriched languages of a lower status. The influence of Latin and Greek (directly or via other languages) on the Germanic languages, for instance, has been enormous and we would today be unable to manage in everyday life without using this "imported" vocabulary. It has become an integral part of the language. No normal user of English finds anything foreign or strange in words like "language", "change", "influence", "special", "vocabulary", "status", or "enter", just to mention a few, all taken from the first two sentences of this essay.

All these words have been nostrified into English and are today normal English words. To what degree the structure of English has changed in the process, and how sudden such changes were, is a question that we cannot discuss in detail here. However, there is some evidence that the transfer from Anglo-Saxon to English was not quite smooth. What we want to analyze in this essay is, how languages

*I am indebted to Helmar G. Frank, who gave me the idea leading up to the basic analytic approach of this essay. Some of the phenomena analyzed here were also discussed in a much simpler framework in German in [1]. I gratefully acknowledge the constructive suggestions of Jens Barthel and Sjur Flåm, which have considerably contributed to improving the quality of the analysis. I had the opportunity to work on this essay during a stay at Fudan University. I thank my host Li Weisen and his colleagues for an interesting and inspiring visit under excellent working conditions. Last, but not least, I am very grateful to Sonja Boden and Judith Wickström for halting and reversing the attrition of my English idiolect in this essay.

manage to import external elements and incorporate these into the structure without causing sudden structural breaks.

We will focus on two aspects of the imported elements, which we will call innovations. Initially, they are introduced by some individuals using them. Through imitation more and more individuals make use of these elements in their social intercourse and the innovations become more and more common. Then the process of absorption into the structure of the language sets in and at the end the innovations are integral parts of the idiom. The diffusion of the intruding elements is modeled as a random-walk stochastic process. The nostrification we see as a Poisson process.

We show that, depending on the parameter values of the model, a language can develop smoothly or be subject to sudden structural changes. We also speculate on the issue which structures are more vulnerable to external intruding elements than others.

The rest of the essay is organized as follows. In section 2, we give a brief overview of some relevant findings in the area of contact linguistics. This is primarily based on some well-known recent standard texts. A formal model is constructed and analyzed in section 3. In the main text, the model is presented in a verbal, non-technical manner, and all technical derivations are relegated to three appendices. Some predictions based on the model are presented in section 4 and the essay closes in section 6 with an outlook.

## 2. LANGUAGE CONTACT

In the literature on contact linguistics, several different phenomena and approaches related to the influence of one language on another are discussed and analyzed.[1] One can divide the analyses of contact linguistics into three broad categories: "borrowing" of both lexical and structural material from one language into another; language shift in bilingual situations; and the emergence of new languages through the fusion of two (or more) languages. A classic example of a language shift is when – in the community of Hungarian speakers in Burgenland – individuals first became bilinguals in Hungarian and German and then monolinguals in German over a few generations, abandoning the use of Hungarian in one domain after the other.[2] The emergence of new languages, we find mainly in the rise of pidgins and creoles.[3]

In this essay, we are concerned with the "borrowing" from other languages. Various terms are used in the literature to discuss changes in the structure of the language system due to the import of elements from other languages: code switching, code mixing, borrowing, transference etc.[4] We will talk about imported elements as externally induced innovations in the lexicon, phonology, morphology, and other aspects of the structure of the language.[5] Our focus is on the temporal process the innovation goes through: it enters the language, is dissipated among the speakers, being initially felt by them to be a "foreign" element which over time is slowly absorbed, "nostrified", into the linguistic system (or rejected and disappearing).

We assume that the flow of foreign elements entering a language and spreading among its users is primarily determined by the sociolinguistic situation, whereas the structure of the affected language to a large extent determines the rate at which the elements are absorbed into it. In this way, we attempt to provide a synthesis of the apparent opposite viewpoints of the determination of the acceptance of foreign material into a language. In order to model the importation in this manner, we need to focus on the time dimension of the import process. This process is made up of two distinct sub-processes. On the one hand, there is the diffusion of the usage of the imported elements in the language commu-

[1] We do not attempt to provide a systematic or representative review of this field here. Only the concepts relevant to our analysis are referred to. For a systematic overview, the reader is referred to a comprehensive treatise on the field of contact linguistics, for instance to [2–4] and the many references therein.

[2] See the study of [5]. A comprehensive treatise on – among other aspects of language change – language shift of immigrants in Australia is [3]. In this area there also exists a number of formal models, notably: [6–11].

[3] A standard text is [12, 13].

[4] See [3], chapter 3.

[5] [2] in chapter 4 discuss the various levels of borrowing as a function of the cultural pressure the dominant language exerts on the recipient one.

nity, and on the other hand, there is the absorption of the imported material into the structure of the language. As it turns out, the exogenous, sociologically determined, influx of new material can – in a dynamic steady-state – be analytically separated from the endogenous, structurally determined, rate of nostrification of the material and analyzed in a very simple model.

In the literature, language change and language shift often go hand-in-hand. The influence of a dominant language forces a minority tongue to leave one domain after the other, leading to attrition and decay as the speakers slowly stop using the language and switch to the dominant one. In this essay, we are primarily interested in how resistant the importing language and its speakers are to such external influences. This resistance we see as determined both by the rate of influx and diffusion (the sociolinguistic aspect) and the rate of absorption (determined both by the linguistic structure and the social rôle of the language). There can, however, be considerable, but stable changes in a language over time without attrition or decay as the well-known examples of the Balkan Sprachbund[6] or Cappadocian Greek[7] demonstrate. By focusing on language change, we are not directly concerned with the shift aspect of the problem. In many cases, where the innovations lead to attrition and decay, however, the shift is implied.[8] This might very well potentially be the most interesting application of this essay. The detailed modeling of the diffusion process is very flexible and can easily be modified to approximate many real-life situations.

### 2.1 Rate of borrowing

In the literature, two main explanations of borrowing are discussed. On the one hand, the social and cultural situation is seen as the most important factor behind the import of features from one language into another, as a rule from a dominant "high-status" language to an idiom of lower social or cultural status.[9] On the other hand, also the structure of the importing language is regarded as a determinant of the ease of import of different linguistic material.[10] This latter aspect, we call absorption. We provide a simple framework where both of these aspects are taken into account and interacting with one another.

We, hence, analytically separate the rate of influx of innovations from their nostrification. It is then natural to model this influx primarily as a function of the relative social and cultural status of the donor and recipient languages.

### 2.2 Diffusion

As already mentioned, the time aspect is very important for our arguments. Hence, the diffusion of an innovation in the population as a function of time is at the core of the model. The classical treatment of the diffusion of innovations in the social-science literature is [16]. In sociolinguistics one of the first models using diffusion methods studies the spread of different sound changes in Chinese syllables that are traced from one type of syllable to another.[11] This author finds the typical *S*-shaped curve with the change first slowly spreading to a few syllables then accelerating and then slowing down again as it affects the last non-affected syllables (or stops before the change is universal). This can be called *W*-diffusion: a certain property is spread from one part of the lexicon to another. The spread from speaker to speaker can be termed *S*-diffusion.[12] This is what we are concerned with.

Most models are driven by an assumption that diffusion occurs by contact and imitation. These models are as a rule deterministic, modeling changes in fractions of users of innovations deterministically.[13] A consequence of this is that an innovation continuously spreads until it is adopted by all potential users.[14]

[6] Cf. [4], chapter 3.

[7] Cf. [4], chapter 7.

[8] Decreolisation, that [14] in chapter 11 refers to as "language suicide ", is an extreme example.

[9] See [15]; [2], chapters 2, 3, and 4; as well as [4], chapter 2.

[10] [4], chapter 2. In addition, there is an "implication table" of the order at which various types of elements are imported as the influence of the exporting language on the importing one grows: first lexicon, then phonology, and, finally, syntax and morphology. See the "borrowing scale" in [2], chapter 4.

[11] See [17].

[12] See [18].

[13] This is also the case in [1].

[14] Some fraction of speakers might be totally resistant and under no circumstances accept the innovation.

The process only goes one way. By modeling the diffusion as a stochastic process, we avoid this problem and the fraction of speakers using the innovation at any time is a stochastic variable taking a random walk, whose expected value is the average number of users.[15] In addition, using a stochastic process, would allow us to model different individual behavior in a more realistic fashion. It is a well-established fact that innovations spread at a different rate in different social groups and cross the borders of different social groups with different propensities.[16] In the stochastic modeling, this is easily accommodated by choosing different adoption and rejection probabilities for different sociologically determined groups of persons and by making the probabilities of encounters between individuals belonging to different groups group dependent.

### 2.3 Absorption

With the absorption of an innovation, we imagine the step from adoption to adaptation.[17] The phenomenon is treated in the literature,[18] but we know of no study investigating the temporal side of this process. Our assumption is that with repetitive usage individuals adapt the innovation phonologically, morphologically etc. until it has become an integral part of the receiving language in the view of its speakers. In the absence of any specific information about the absorption process, we make the simplest possible assumption about the expected time it takes for an innovation to be absorbed: it is directly determined by the frequency of its usage. That is, at each encounter involving the innovation there is a certain given probability that it will be adapted to the structure of the receiving language. The speed of this process is supposed to capture the various structurally determined constraints on the import of external elements.

We speculate that the resistance to adaptation is both a sociolinguistic issue and a matter of the structure of the receiving language. The greater the number of steps (phonological, morphological etc.) an innovation has to go through to be integrated into the language, the slower the adaptation process is assumed to be.

## 3. THE FORMAL MODEL

We first model how an intrusion or an innovation dissipates through society. At any time, for a given innovation, there are three types of individuals in the language community. $F$ persons use the innovation, but consider it a foreign element in their language; $A$ individuals use the innovation and consider it an integral part of the language; and $R$ persons do not use it. With $P$ we denote the sum of $F$ and $A$ and $N$ is the total size of the language community:

$$N = P + R = F + A + R \tag{3.1}$$

For the sake of analysis, we assume that the contacts between the individuals occur pairwise and are consecutively numbered by $\theta$. By assuming that there is a fixed number of transactions per unit of time, we will transfer the model into continuous time. At each encounter, the individuals mutate between the three groups with certain probabilities. Per unit of time, an exogenously determined number of innovations enter the language. We define the heterogeneity of the language as the sum of the number of individuals using an innovation without considering it an integral part of the idiom, *i. e.*, the sum of the $F$ over all innovations.

We look for steady states of this system.

[15] See figures 1 and 2.
[16] See [15].
[17] See [14], chapter 8.
[18] See, for instance, [4], chapter 2.

### 3.1 Spread of innovations

An innovation – a new word, say – is assumed to enter the language from outside creating an $F$ individual.[19] This individual interacts with other speakers and the innovation is then adopted with a certain probability by such a person after an encounter. Specifically, an $RR$ encounter does not influence the spread of the innovation; a $PP$ encounter does not influence the spread of the innovation, but it can influence the nostrification, *i. e.* the relative sizes of $A$ and $F$. A $PR$ encounter, on the other hand, influences the spread of the innovation: with probability $\alpha$, the $R$ individual mutates into a $P$ ($F$ or $A$) individual and with probability $\beta$, the $P$ individual mutates into an $R$ individual; with probability $(1-\alpha-\beta)$, no mutation takes place. We further define $\gamma$ as $\beta/\alpha$.

That is, the spread of innovations is due to imitation. In appendix A the dynamics of the probability density of the distribution of the $P$'s, $\delta^P(\theta)$, and the size of the expected value of $P$, $\bar{P}$, are found, as well as the fraction $\bar{p} := \bar{P}/N$:

$$\bar{P}(\theta+1) - \bar{P}(\theta) = 2\alpha(1-\gamma)\left[\bar{P}(\theta)\,\frac{N-\bar{P}(\theta)}{(N-1)\,N} - \frac{\sigma_P^2(\theta)}{(N-1)\,N}\right] \tag{3.2}$$

and, as a function of the "age" $\tau$ of the innovation in the language:

$$\dot{\bar{p}}(\tau) = 2\omega\alpha(1-\gamma)\,\frac{N}{N-1}\left\{[1-\bar{p}(\tau)]\,\bar{p}(\tau) - \sigma_p^2(\tau)\right\} \tag{3.3}$$

The parameter $\omega$ is the number of encounters per unit of time and number of individuals in the population.

It is clear that $\bar{p}(\tau)$ has the expected $S$-form. Figure 1 shows the probability density of $\bar{P}$ for different values of $\theta$. Here, we have set $N = 10, \alpha = 0.075$, and $\beta = 0.0075$. It is interesting to note that the mass of the probability density is concentrated at the lower end as well as at the higher end of the distribution. With time the concentration at the higher end increases and at the lower end decreases. That is, a typical innovation is either used by a few people or disappears or is, after a short time, used by virtually everyone. This is in agreement with the general findings in diffusion analysis; an innovation spreads slowly in the initial phase, then very rapidly in a middle phase, and at the end of the diffusion process it reaches the last potential users very slowly.[20] In figure 2 the expected number of users of the innovation, *i. e.*, the resulting average dissipation of an innovation as a function of the number of encounters is depicted for different values of $\alpha$ and $\beta$.

### 3.2 Absorption

The absorption is modeled as a spontaneous mutation of an individual who has adopted the innovation into an individual who has adopted it and for whom it is no more a foreign element of the language.[21] If the probability of such a mutation of an $F$ individual in any period is given by $\lambda$ and the number of individuals having adopted the innovation at the beginning of period $\theta$ with probability $\delta^P(\theta)$ is $P(\theta) = F^*(\theta) + A^*(\theta)$, then at the end of the period the expected number of $F(\theta)$ is $(1-\lambda)F^*(\theta)$ and the expected number of $A$ is $A^*(\theta) + \lambda F^*(\theta)$. In other words, since the absorption process is stochastically independent of the diffusion process, the expected value of $F$ can be written as:

$$\bar{F}(\theta) = \sum_{P=0}^{N} \delta^P(\theta)(1-\lambda)^\theta P = (1-\lambda)^\theta \sum_{P=0}^{N} \delta^P(\theta) P = (1-\lambda)^\theta \bar{P}(\theta) \tag{3.4}$$

[19] Of course, there are also innovations from the "inside". These are part of the system of the language, however, and do not need to be absorbed into the language system. They start as an $A$ individual and can be assumed to spread in the same way as external innovations.

[20] See [16].

[21] This leads to a Poisson process. Such processes are used, for instance, to model nuclear decay.

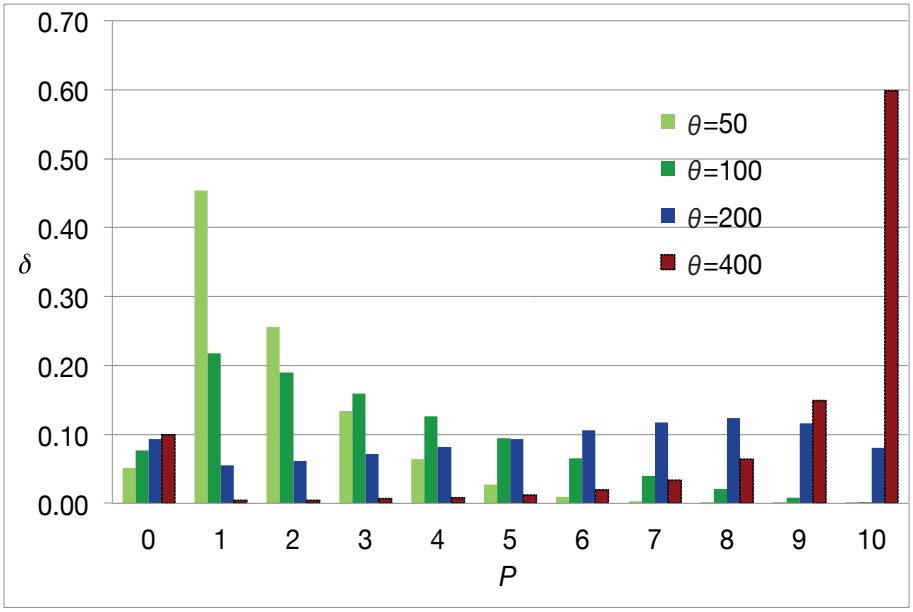



**FIGURE 1:** Probability densities of the diffusion of an innovation

This is the expected number of users of the innovation who consider it a foreign element. In figure 3 the expected value of $\bar{F}$ as a function of $\theta$ is shown for $\lambda = 0.002$ and the same values of $\alpha$ and $\beta$ as in figure 2.

If $\lambda$ is not stationary, but changes due to external influences from one encounter to the next, we have to number the encounters independently of the "encounter age" $\theta$ of each individual innovation. This absolute numbering of encounters we denote by $\eta$ and equation 3.4 will then be written as:

$$\bar{F}(\theta,\eta) = \prod_{i=1}^{\theta} [1-\lambda(\eta-i+1)] \sum_{P=0}^{N} \delta^P(\theta) P \tag{3.5}$$

With a suitable choice of units, we write the expected fraction of the population using the innovation and considering it a foreign element as a function of time $t$ and the "age" $\tau$ of the innovation in the language, $\bar{f}(\tau,t)$:

$$\bar{f}(\tau,t) = e^{Q(\tau,t)}\bar{p}(\tau) \tag{3.6}$$

The function $Q$ is given by:

$$Q(\tau,t) := \rho \int_0^{\tau} \ln[1-\lambda(t-\kappa)]\, d\kappa \tag{3.7}$$

The positive constant $\rho$ in this expression is related to the number of encounters per unit of time.

### 3.3 Heterogeneity

We will call $\bar{f}$ the contribution of this innovation to the heterogeneity of the language. Since there is a steady stream of innovations at the rate $n(t)$ entering the language over time, the total heterogeneity at any time $t$, $H(t)$, can be defined as:

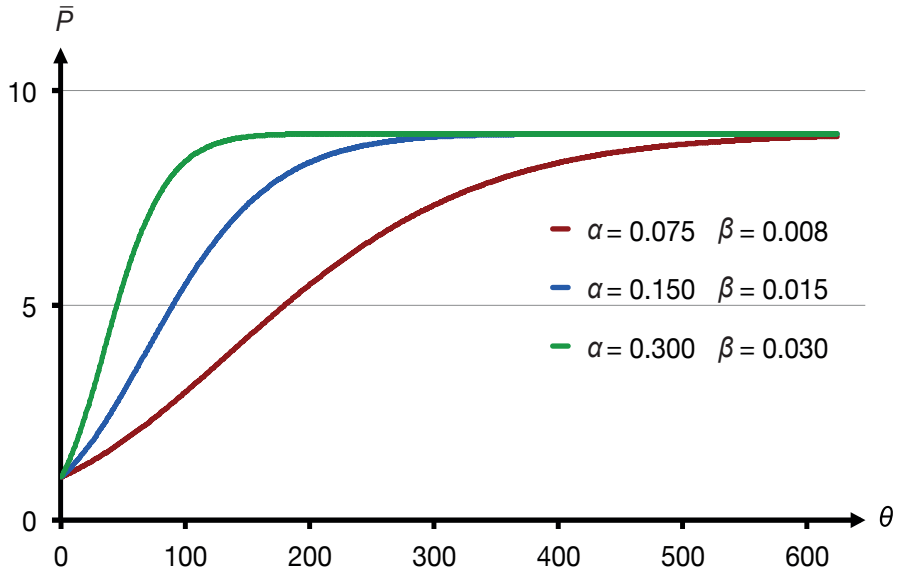



**FIGURE 2:** Average dissipation of an innovation

$$H(t) := \int_0^\infty n(t-\tau)\bar{f}(\tau,t)d\tau = \int_0^\infty n(t-\tau)\,e^{Q(\tau,t)}\bar{p}(\tau)d\tau \tag{3.8}$$

We assume that the rate of absorption depends on the heterogeneity. That is, a very heterogeneous language has a lower rate of absorption than a homogeneous language:

$$\lambda(t) = \tilde{\lambda}\left[H(t)\right], \quad \frac{\partial\tilde{\lambda}}{\partial H} \leq 0 \tag{3.9}$$

Substituting $\omega$ for $t-\tau$, we find:

$$H(t) = \int_{-\infty}^{t} n(\omega)\,e^{Q(t-\omega,t)}\bar{p}(t-\omega)d\omega \tag{3.10}$$

We have in 3.10 an integral equation for the heterogeneity of the language. In the following, we will characterize its solution for an exogenous stream of innovations $n(t)$.

### 3.4 The dynamics of the heterogeneity

In appendix B it is shown that the dynamics of $H$ can be expressed by:

$$\begin{aligned}\dot{H} &= \rho\ln\left\{1-\tilde{\lambda}\left[H(t)\right]\right\}H(t) - n(t)\rho\int_0^\infty \ln\left\{1-\tilde{\lambda}\left[H(t-\tau)\right]\right\}e^{Q(\tau,t)}\bar{p}(\tau)d\tau \\ &+ \int_0^\infty\left[n(t-\tau)\left(Q_2+\rho\ln\frac{1-\tilde{\lambda}\left[H(t-\tau)\right]}{1-\tilde{\lambda}\left[H(t)\right]}\right)+n'(t-\tau)\right. \\ &\left.+\rho\left[n(t)-n(t-\tau)\right]\ln\left\{1-\tilde{\lambda}\left[H(t-\tau)\right]\right\}\right]e^{Q(\tau,t)}\bar{p}(\tau)d\tau\end{aligned} \tag{3.11}$$

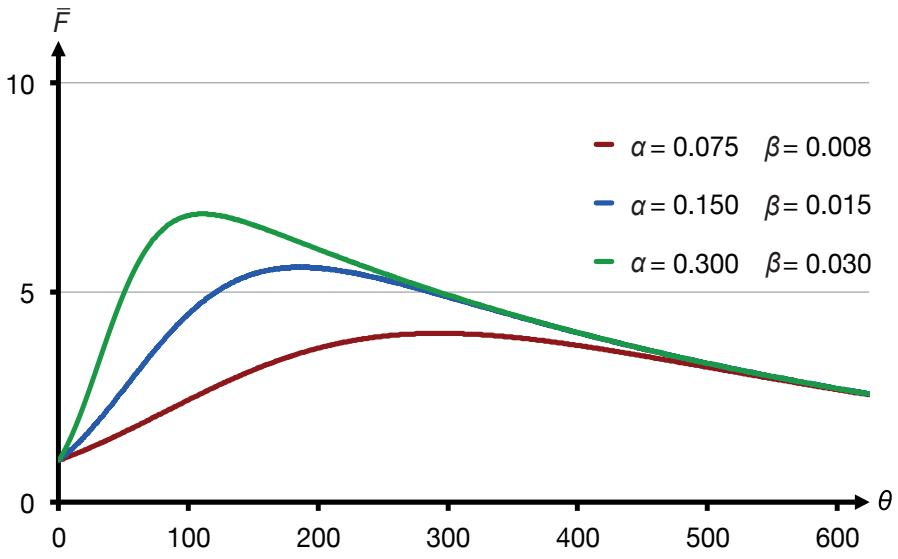



**FIGURE 3:** Expected number of users of a non-absorbed innovation with $\lambda = 0.002$

This expression is quite complicated and ultimately determined by the history of $n$. In other words, we would have to use the exogenous path of $n$ to find the path of $H$ from some initial value, $H_0$, with the help of this equation. To make the analysis tractable we will, however, limit ourselves to a comparison of the long-term steady states.

### 3.5 Steady state

In a steady state, $n$ and, consequently, $H$ are stationary. The condition for a sustainable steady state is then:

$$\ln\left[1-\tilde{\lambda}(H)\right] H = n\int_0^{\infty} \ln\left[1-\tilde{\lambda}(H)\right] e^{\tilde{Q}(\tau,H)}\bar{p}(\tau)d\tau \tag{3.12}$$

Here, $\tilde{Q}(\tau, H)$ is defined by:

$$\begin{aligned}\tilde{Q}(\tau,H) &:= \rho\int_0^{\tau} \ln\left[1-\tilde{\lambda}(H)\right] d\kappa \\ &= \rho\tau\ln\left[1-\tilde{\lambda}(H)\right]\end{aligned} \tag{3.13}$$

In appendix C it is shown that the steady-state condition reduces to:

$$n = g(H) \tag{3.14}$$

The function $g(H)$ is defined in appendix C and describes the long-run capacity of the language to absorb innovations without increasing the heterogeneity. It will – under our assumptions – have the general form of figure 4.

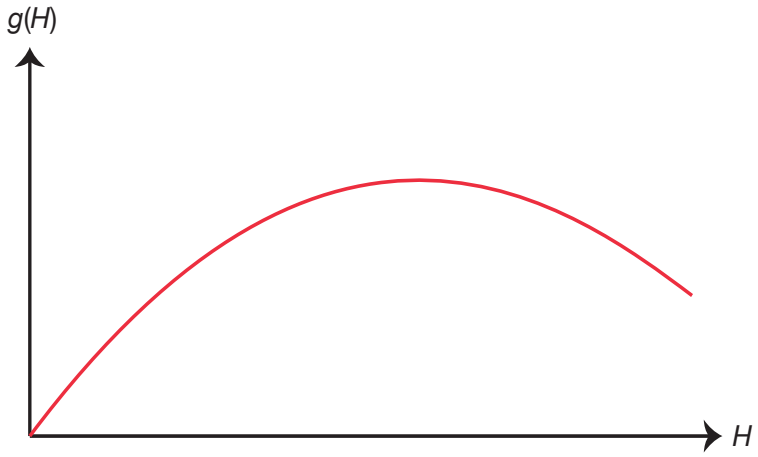



**FIGURE 4:** The absorption function

### 3.6 Dynamic equilibria

Not every steady state is a stable dynamic equilibrium. Intuitively, it is clear that if $n$ exceeds $g(H)$, $H$ will increase and inversely if $n$ is smaller than $g(H)$. To show this in a stringent manner, we observe that for a constant $n$ equation 3.12 takes the form:

$$\dot{H} = \rho \ln \left\{ 1 - \tilde{\lambda}[H(t)] \right\} H(t) + n \int_0^{\infty} \left[ Q_2 - \rho \ln \left\{ 1 - \tilde{\lambda}[H(t)] \right\} \right] e^{Q(\tau,t)} \bar{p}(\tau) d\tau \qquad (3.15)$$

This equation relates the rate of change in $H$ to the current value of $H$ as well as to its history captured in $Q$. We note that $Q_2$ takes the sign of the rate of change in $H$ and in a steady state is equal to zero. It reacts with a delay to changes in $H$, and a perturbation in $H$ from a steady state will initially have a negligible influence on $Q_2$, but its value will change as time goes on if the value of $H$ changes over time; becoming positive and growing if $H$ grows and the opposite if $H$ decreases.

The absorption function allows for two types of steady states, see figure 5. Now assume that there is a small perturbation in $H$ moving it away from point $A$. An increase in $H$ will make $n$ smaller than $g(H)$, and the right-hand side of equation 3.15 becomes negative; $H$ will decrease and move back towards point $A$.[22] A positive perturbation away from point $B$ will make $n$ greater than $g(H)$, and the right-hand side of equation 3.15 becomes positive; $H$ will continue to grow and with time also $Q_2$ will become positive and grow, enforcing the growth rate of $H$. The system is unstable.

The corresponding results are obtained for a negative perturbation in $H$. A move away from point $A$ will make the right-hand side of equation 3.15 positive, and $H$ will return towards point $A$. A negative perturbation away from point $B$ will make the rate of change negative, and $H$ will continue moving away from point $B$. With time, $Q_2$ becomes negative and will be growing in absolute value. This will reinforce the motion away from $B$, and the decreasing $H$ will overshoot point $A$ before changing direction. The system will eventually came to a rest in point $A$ after oscillating around this point due to the delayed reactions captured by $Q$.[23]

## 4. COMPARATIVE ANALYSIS

The model above provides us with a tool to discuss which languages might be threatened by structural decay and which will be dynamically stable. We hint at a classification of different languages (largely

[22] We ignore the effect of $Q_2$, which will introduce some oscillating behavior around the point $A$. See the discussion below.

[23] Formally, one cannot exclude an ever stronger amplitude of these oscillations without specifying the limits on the dependency of $\lambda$ on $H$. Such an exploding behavior, however, can in any sensible specification of this functional relationship be excluded.

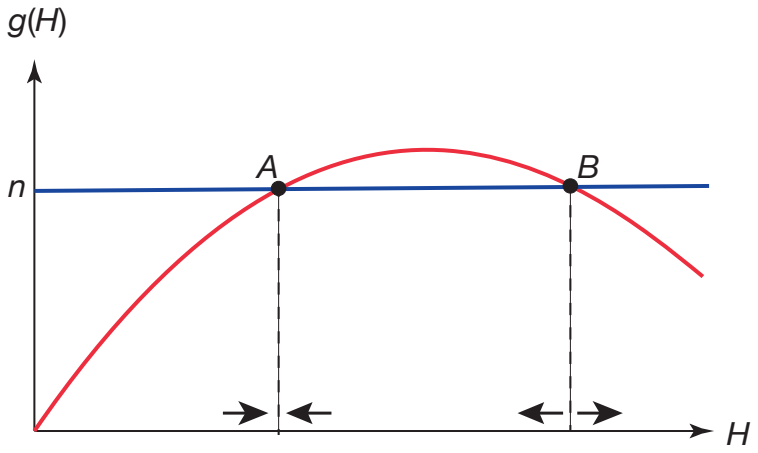



**FIGURE 5:** Possible steady states

based on anecdotal evidence) according to their structural types and degree of normalization, as well as cultural and social status.[24]

There are, in essence, two parameters of the model that are crucial for our analysis: the rate of import of innovations, $n$, and the rate of nostrification, captured by $\lambda$. As mentioned above, the rate of import is assumed to be mainly a result of the (relative) status of the language, whereas the nostrification rate is taken to be determined by the structure and degree of normalization. The interaction of these aspects is analyzed in a simple diagram comparing possible long-run steady-state equilibria. Finally, some (very speculative) policy implications will be drawn.

### 4.1 Rate of innovation

The rate of innovation, $n(t)$, is assumed to be exogenously given and to depend primarily on the relative cultural and social status of the donor and recipient languages, as we noted above in section 2.1. The question here is how this relationship is determined by other factors and how it can be altered through a conscious language policy.

An external factor that has become increasingly important in recent years is globalization, be it due to expanding trade, the increased spread of culture from one land to another through new media and reduced transaction costs, or easy direct access to individuals all over the world with the help of the internet. Especially the accelerating dominance of (the American variant of) English in many international domains has lead to an increased borrowing from American of both vocabulary and structural elements in virtually any language of the world.

Some countries, like France or Iceland, try to counteract this borrowing with corpus planning. The degree of success seems to be variable. If the rate of borrowing in a minority language from the majority tongue depends on the relative status of the two idioms, the obvious way to influence the borrowing rate is through status planning. Giving the minority language some official status would presumably also increase its cultural and social status. Also corpus planning, however, could have an influence here.

### 4.2 Language structure

As noted in section 2, the structure of the language might not directly influence the borrowing, but could affect the rate of nostrification. Languages that are similar might more easily incorporate elements from one another, than languages that are far apart. Also, adapting an imported verb, say, into an isolating language like Chinese, might be easier, basically only requiring a phonological adaptation, than adapting an imported verb into a highly inflected language like Russian, where in addition to phonetics also a

[24] A further discussion of the various classification possibilities can be found in [1].

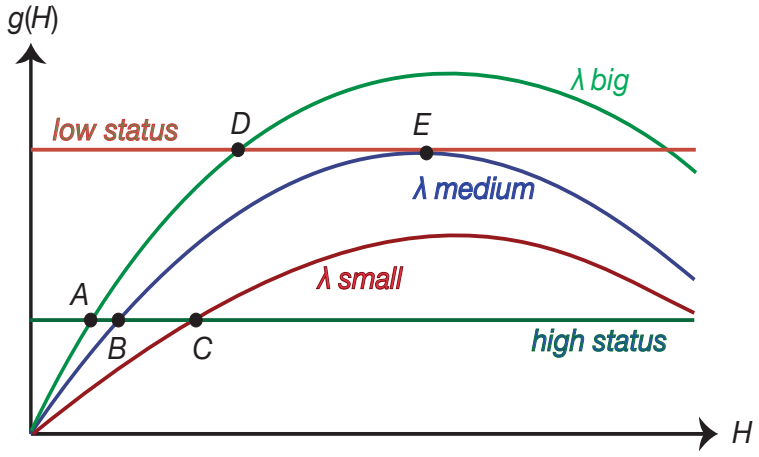



**FIGURE 6:** Possible stability scenarios

considerable adaptation to a rich set of conjugation forms is necessary.

Adaptation might also be facilitated by clear rules. Simplifying, we could say that codified written languages more rapidly absorb imported elements than languages with mainly an oral tradition. That is, nostrification might come faster in languages with a long written tradition than in languages that are used primarily orally.

### 4.3 Structural instability

The previous discussion can be summarized in figure 6. We see that languages possessing a stabilized written form, which are conjectured to have a high $\lambda$, can be assumed to have a stable structure, a long-run equilibrium in points *A* or *D*, depending on their status. On the other hand, a language without a written codification could be border-line unstable or unstable if it has a low status, point *E*. Many creole languages seem to fit this image. Jamaican Creole or Hawai'ian Creole seem to be unstable going through a process of decreolization. Other creole languages, like Tok Pisin, Bislama or Haitian, on the other hand, seem to stabilize due to a higher status as official languages, point *B*.

### 4.4 Policy implications

The policy conclusions that can be drawn from this seem to be that increasing the status of a language, for instance giving it an official status, might stabilize it, moving it from *E* to *B* or from free fall to *C*. Also corpus planning, providing a written norm might help, inducing a movement from free fall to *E*, or from *E* to *D*.

## 5. OUTLOOK AND EXTENSIONS

As mentioned before, the detailed modeling of the diffusion process can be extended to include more complex social structures. One can define different social groups with different contact probabilities between individuals in the group and with individuals outside the group. The adoption probabilities of innovations can differ for different individuals and groups. This type of analysis can easily be accomplished in a computational version of the model, wherein real-life situations can be approximated and simulated.[25]

A next step would, hence, be to implement a computational version of the model and make simulations. This type of simulations could prove to be a valuable method in analyzing language death through attrition, a phenomenon that threatens a considerable portion of the world's 6000 or so languages. To understand the individual processes leading to this attrition, might not be a sufficient condition for reversing the process of language death in most cases, but it might well be a necessary condition.

[25] [15] describes many such stratified situations.

## 6. CONCLUDING REMARK

There is an extensive amount of literature discussing language death in terms of language shift. With this essay we try to focus on language death through unstable structures. In many cases the two effects go hand-in-hand. Language shift lowers the status of a language and as a consequence might make the structure unstable. The details of these processes are largely unknown, though. It is especially the different rates of change that have not been extensively studied.

Many arguments in this essay are to a large extent rather speculative and intuitive. We know a bit about how languages adapt innovations and make them part of the system. However, we know very little about how fast the adaptation processes are and what determines their speed. We can only make some general assumptions based on anecdotal evidence or introspection.

What we have attempted to demonstrate, though, is that due to the interaction of the process of adopting external innovations with the process of their internal adaptation, the issue of time and the relative velocity of these processes are of considerable importance in the analysis of language attrition and decay, and consequently of language shift and death. If the structural properties of languages are important for these rates of adjustment, then we cannot ignore the structure in analyzing language shift.

## 7. REFERENCES

[1] B.-A. Wickström. "Die Sprache als erneuerbare Ressource: Die Kapazität verschiedener Sprachen, fremde Elemente zu nostrifizieren." In: *Florilegium Interlinguisticum: Festschrift für Detlev Blanke zum 70. Geburtstag*. Ed. by C. Brosch and S. Fiedler. Frankfurt am Main: Peter Lang, 2011, pp. 193–208.

[2] S. G. Thomason and T. Kaufman. *Language Contact, Creolization and Genetic Linguistics*. Berkeley: University of California Press, 1988.

[3] M. Clyne. *Dynamics of Language Contact: English and Immigrant Languages*. Cambridge Approaches to Language Contact. Cambridge: Cambridge University Press, 2003.

[4] D. Winford. *An Introduction to Contact Linguistics*. Vol. 33. Language in Society. Oxford: Blackwell, 2003.

[5] S. Gal. *Language Shift: Social Determinants of Linguistic Change in Bilingual Austria*. New York: Academic Press, 1979.

[6] D. M. Abrams and S. H. Strogatz. "Modelling the dynamics of language death." *Nature*, vol. 424, p. 900, 2003.

[7] J. W. Minett and W. S.-Y. Wang. "Modelling endangered languages: The effects of bilingualism and social structure." *Lingua*, vol. 118, pp. 19–45, 2008.

[8] B.-A. Wickström. "Can bilingualism be dynamically stable? A simple model of language choice." *Rationality and Society*, vol. 17, no. 1, pp. 81–115, 2005.

[9] C. Fernando, R.-L. Valijärvi, and R. A. Goldstein. "A model of the mechanisms of language extinction and revitalization strategies to save endangered languages." *Human Biology*, vol. 82, no. 1, pp. 47–75, Feb. 2010.

[10] M. Patriarca and T. Leppänen. "Modeling language competition." *Physica A*, vol. 338, pp. 296–299, 2004.

[11] X. Castelló, L. Loureiro-Porto, V. M. Eguíluz, and M. San Miguel. "The fate of bilingualism in a model of language competition." In: *Advancing Social Simulation: The First World Conference*. Ed. by S. Takahashi, J. Sallach, and Rouchier. Tokyo: Springer-Verlag, 2007, pp. 83–94.

[12] J. A. Holm. *Pidgins and Creoles: Volume 1, Theory and Structure*. Cambridge Language Surveys. Cambridge: Cambridge University Press, 1988.

[13] J. A. Holm. *Pidgins and Creoles: Volume 2, Reference Survey*. Cambridge Language Surveys. Cambridge: Cambridge University Press, 1989.

[14] A. M. S. McMahon. *Understanding Language Change*. Cambridge: Cambridge University Press, 1994.

[15] W. Labov. *Principles of Lingustic Change, Volume 2: Social Factors*. Vol. 29. Language in Society. Oxford: Blackwell, 2001.

[16] E. M. Rogers. *Diffusion of Innovations*. 5th ed. New York: The Free Press, 2003. (orig. pub. as: *Diffusion of Innovations*. New York: The Free Press, 1962.)

[17] M. Chen. "The time dimension: Contribution toward a theory of sound change." *Foundations of Language*, vol. 8, no. 4, pp. 457–498, Jul. 1972.

[18] W. S.-Y. Wang and J. W. Minett. "The invasion of language: Emergence, change and death." *Trends in Ecology and Evolution*, vol. 20, no. 5, pp. 263–269, 2005.

# APPENDICES

## A. THE DISPERSION FUNCTION

The dispersion of an innovation is a stochastic process. We assume that there is one random encounter in each period. The periods are denoted by $\theta$. The number of users of the innovation at the beginning of encounter $\theta$, $\tilde{P}(\theta)$, is a stochastic variable, which is realized as $P(\theta) \in \{0, 1, 2, ..., N\}$. Let the probability density be $\delta^P(\theta)$. The probability that the number of users of the innovation changes from encounter number $\theta$ to encounter number $\theta + 1$ depends on whether in the encounters of two people, one is a user of the innovation and the other isn't. Let the probability that after such an encounter both use the innovation be $\alpha$ and the probability that neither use it $\beta$. If $P$ persons use the innovation and $N - P$ do not, the probability that the next encounter is of this type, is given by:

$$\xi(P) = 2\frac{(N-P)\,P}{(N-1)\,N} \tag{A.1}$$

The number of users will then increase by one person with probability $\xi\alpha$, decrease by one person with probability $\xi\beta$, and remain constant with probability $1 - (\alpha + \beta)\,\xi =: 1 - \alpha\,(1 + \gamma)\,\xi$, where $\gamma$ is defined as $\beta/\alpha$.

We can now find the probability density $\delta^P(\theta + 1)$ after encounter $\theta + 1$. $P = 0$ can occur in two ways: If $P$ in period $\theta$ were zero, $P$ stays equal to zero; if $P$ were one and this person gives up the usage, $P$ becomes zero which happens with probability $\xi(1)\beta$. Hence:

$$\delta^0(\theta + 1) = \delta^0(\theta) + \alpha\gamma\xi(1)\delta^1(\theta) \tag{A.2}$$

or

$$\delta^0(\theta + 1) - \delta^0(\theta) = \alpha\gamma\xi(1)\delta^1(\theta) \tag{A.3}$$

Similarly, we find:

$$\delta^1(\theta+1) = [1-\alpha(1+\gamma)\,\xi(1)]\,\delta^1(\theta) + \alpha\gamma\xi(2)\delta^2(\theta) \tag{A.4}$$

or

$$\delta^1(\theta+1) - \delta^1(\theta) = \alpha\left[-(1+\gamma)\,\xi(1)\,\delta^1(\theta) + \gamma\xi(2)\delta^2(\theta)\right] \tag{A.5}$$

For $2 \leq P \leq N-2$, the expression becomes:

$$\delta^P(\theta+1) = [1-\alpha(1+\gamma)\,\xi(P)]\,\delta^P(\theta) + \alpha\gamma\xi(P+1)\delta^{P+1}(\theta) + \alpha\xi(P-1)\delta^{P-1}(\theta) \tag{A.6}$$

or

$$\delta^P(\theta+1) - \delta^P(\theta) = \alpha\left[\xi(P-1)\delta^{P-1}(\theta) - (1+\gamma)\,\xi(P)\,\delta^P(\theta) + \gamma\xi(P+1)\delta^{P+1}(\theta)\right] \tag{A.7}$$

For $P = N-1$, we have:

$$\delta^{N-1}(\theta+1) = [1-\alpha(1+\gamma)\,\xi(N-1)]\,\delta^{N-1}(\theta) + \alpha\xi(N-2)\delta^{N-2}(\theta) \tag{A.8}$$

or

$$\delta^{N-1}(\theta+1) - \delta^{N-1}(\theta) = \alpha\left[\xi(N-2)\delta^{N-2}(\theta) - (1+\gamma)\,\xi(N-1)\,\delta^{N-1}(\theta)\right] \tag{A.9}$$

Finally, for $P = N$, the expression is:

$$\delta^N(\theta+1) = \delta^N(\theta) + \alpha\xi(N-1)\delta^{N-1}(\theta) \tag{A.10}$$

or

$$\delta^N(\theta+1) - \delta^N(\theta) = \alpha\xi(N-1)\delta^{N-1}(\theta) \tag{A.11}$$

We note that the system of difference equations is scaled by $\alpha$. Hence, an increase in $\alpha$ by constant $\gamma$ will make the process go faster, but will in no other way influence it. The parameter $\gamma$ will determine how many innovations survive in the end. In order to find the success rate of innovations, we combine the equations above substituting each one into the next for increasing values of $\theta$, to find comparable expressions for all the $\delta$'s.

Since $\delta^0(0) = 0$, we find from A.2:

$$\delta^0(\theta) = \alpha\gamma\xi(1)\left[\sum_{\tau=0}^{\theta-1}\delta^1(\tau)\right] \tag{A.12}$$

Similarly for $\delta^1$, noting that $\delta^1(0) = 1$:

$$\delta^1(\theta) = 1 + \alpha\left[-(1+\gamma)\,\xi(1)\sum_{\tau=0}^{\theta-1}\delta^1(\tau) + \gamma\xi(2)\sum_{\tau=0}^{\theta-1}\delta^2(\tau)\right] \tag{A.13}$$

In general, for $2 \leq P \leq N-2$, since $\delta^P(0) = 0$, we find:

$$\delta^P(\theta) = \alpha\left[\xi(P-1)\sum_{\tau=0}^{\theta-1}\delta^{P-1}(\tau) - (1+\gamma)\,\xi(P)\sum_{\tau=0}^{\theta-1}\delta^P(\tau) + \gamma\xi(P+1)\sum_{\tau=0}^{\theta-1}\delta^{P+1}(\tau)\right] \tag{A.14}$$

for $P = N - 1$, the result is:

$$\delta^{N-1}(\theta) = \alpha \left[ \xi(N-2) \sum_{\tau=0}^{\theta-1} \delta^{N-2}(\tau) - (1+\gamma)\, \xi(N-1) \sum_{\tau=0}^{\theta-1} \delta^{N-1}(\tau) \right] \tag{A.15}$$

and finally for $P = N$, the expression becomes:

$$\delta^{N}(\theta) = \alpha \xi(N-1) \sum_{\tau=0}^{\theta-1} \delta^{N-1}(\tau) \tag{A.16}$$

As $\theta \to \infty$, the limiting values are:

$$\begin{array}{lcll} \delta^0(\theta) & \to & \delta^0 \geq 0 & \\ \delta^P(\theta) & \to & 0, & 1 \leq P \leq N-1 \\ \delta^N(\theta) & \to & \delta^N = 1 - \delta^0 \geq 0 & \end{array} \tag{A.17}$$

Using this fact and defining $\Delta(P) := \xi(P) \sum_{\tau=0}^{\infty} \delta^P(\tau)$, we rewrite equations A.12 through A.16 as:

$$\delta^0 = \alpha\gamma\Delta(1) \tag{A.18}$$

$$\alpha\,(1+\gamma)\,\Delta(1) = 1 + \alpha\gamma\Delta(2) \tag{A.19}$$

$$\Delta(P-1) = (1+\gamma)\,\Delta(P) - \gamma\Delta(P+1) \tag{A.20}$$

$$\Delta(N-2) = (1+\gamma)\,\Delta(N-1) \tag{A.21}$$

$$\alpha\Delta(N-1) = 1 - \delta^0 \tag{A.22}$$

We first substitute equation A.19 into equation A.18:

$$\delta^0 = \frac{\gamma}{1+\gamma} + \frac{\alpha\gamma^2}{1+\gamma}\Delta(2) \tag{A.23}$$

and then find $\Delta(2)$ from A.20:

$$
\begin{aligned}
\Delta(2) &= (1+\gamma)\,\Delta(3) - \gamma\Delta(4) \\
&= (1+\gamma)\,[(1+\gamma)\,\Delta(4) - \gamma\Delta(5)] - \gamma\Delta(4) \\
&= \left(1+\gamma+\gamma^2\right)\Delta(4) - (1+\gamma)\,\gamma\Delta(5) \\
&= \left(1+\gamma+\gamma^2+\gamma^3\right)\Delta(5) - \left(1+\gamma+\gamma^2\right)\gamma\Delta(6) \\
&= \sum_{i=0}^{P-2}\gamma^i\Delta(P) - \sum_{i=1}^{P-2}\gamma^i\Delta(P+1) \\
&= \sum_{i=0}^{N-4}\gamma^i\Delta(N-2) - \sum_{i=1}^{N-4}\gamma^i\Delta(N-1) \\
&= \left[(1+\gamma)\sum_{i=0}^{N-4}\gamma^i - \sum_{i=1}^{N-4}\gamma^i\right]\Delta(N-1) \\
&= \frac{1-\delta^0}{\alpha}\sum_{i=0}^{N-3}\gamma^i \\
&= \frac{1-\gamma^{N-2}}{1-\gamma}\frac{1-\delta^0}{\alpha}
\end{aligned}
\tag{A.24}
$$

Substituting this into A.23 and solving, we finally arrive at the value of $\delta^0$:

$$
\begin{aligned}
\delta^0 &= \frac{\gamma}{1+\gamma} + \frac{\gamma^2}{1+\gamma}\frac{1-\gamma^{N-2}}{1-\gamma}\left(1-\delta^0\right) \\
\delta^0 &= \gamma\frac{1-\gamma^{N-1}}{1-\gamma^N} = \gamma - \frac{1-\gamma}{1-\gamma^N}\gamma^N
\end{aligned}
\tag{A.25}
$$

For a large $N$ and $\gamma < 1$ this, of course, reduces to:

$$\delta^0 = \gamma \tag{A.26}$$

That is, a fraction $\gamma$ of the innovations does not survive in the long run.

It is of some interest to know how the expected value $\bar{P}$ of $\tilde{P}$ changes with time:

$$
\begin{aligned}
\bar{P}(\theta+1) &:= \sum_{P=0}^{N}\delta^P(\theta+1)P \\
&= \delta^1(\theta) - \alpha\,(1+\gamma)\,\xi(1)\,\delta^1(\theta) + \alpha\gamma\xi(2)\delta^2(\theta) \\
&+ \sum_{P=2}^{N-2}\left[\delta^P(\theta)P - \alpha\,(1+\gamma)\,\xi(P)\,\delta^P(\theta)P\right. \\
&+ \left.\alpha\gamma\xi(P+1)\,\delta^{P+1}(\theta)P + \alpha\xi(P-1)\,\delta^{P-1}(\theta)P\right] \\
&+ \delta^{N-1}(\theta)\,(N-1) - \alpha\,(1+\gamma)\,\xi(N-1)\,\delta^{N-1}(\theta)\,(N-1) \\
&+ \alpha\xi(N-2)\,\delta^{N-2}(\theta)\,(N-1) \\
&+ \delta^N(\theta)N + \alpha\xi(N-1)\,\delta^{N-1}(\theta)N
\end{aligned}
\tag{A.27}
$$

or:

$$\begin{aligned}
\bar{P}(\theta+1) &= \bar{P}(\theta) \\
&- \alpha(1+\gamma)\left[\xi(1)\,\delta^1(\theta)+\sum_{P=2}^{N-2}\xi(P)\,\delta^P(\theta)P+\xi(N-1)\,\delta^{N-1}(\theta)\,(N-1)\right] \\
&+ \alpha\gamma\left[\xi(1)\,\delta^1(\theta)+\xi(2)\delta^2(\theta)2+\sum_{P=2}^{N-2}\xi(P+1)\delta^{P+1}(\theta)\,(P+1)\right] \\
&- \alpha\gamma\left[\xi(1)\,\delta^1(\theta)+\xi(2)\delta^2(\theta)+\sum_{P=2}^{N-2}\xi(P+1)\delta^{P+1}(\theta)\right] \qquad \text{(A.28)} \\
&+ \alpha\left[\sum_{P=2}^{N-2}\xi(P-1)\delta^{P-1}(\theta)\,(P-1)+\xi(N-2)\delta^{N-2}(\theta)\,(N-2)+\xi(N-1)\delta^{N-1}(\theta)\,(N-1)\right] \\
&+ \alpha\left[\sum_{P=2}^{N-2}\xi(P-1)\delta^{P-1}(\theta)+\xi(N-2)\delta^{N-2}(\theta)+\xi(N-1)\delta^{N-1}(\theta)\right]
\end{aligned}$$

Hence, using the fact that $\xi(N)=0$, one finds:

$$\begin{aligned}
\bar{P}(\theta+1) &= \bar{P}(\theta) \\
&+ [\alpha+\alpha\gamma-\alpha(1+\gamma)]\sum_{P=0}^{N}\xi(P)\delta^P(\theta)P \qquad \text{(A.29)} \\
&+ (\alpha-\alpha\gamma)\sum_{P=0}^{N}\xi(P)\delta^P(\theta)
\end{aligned}$$

This gives us:

$$\bar{P}(\theta+1)-\bar{P}(\theta)=\alpha(1-\gamma)\sum_{P=0}^{N}\xi(P)\delta^P(\theta) \qquad \text{(A.30)}$$

Substituting for $\xi$, we find:

$$\begin{aligned}
\sum_{P=0}^{N}\xi(P)\delta^P(\theta) &= \frac{2}{(N-1)\,N}\sum_{P=0}^{N}\delta^P(\theta)\,(N-P)\,P \\
&= \frac{2}{(N-1)\,N}\sum_{P=0}^{N}\delta^P(\theta)\,(NP-P^2) \\
&= \frac{2}{(N-1)\,N}\sum_{P=0}^{N}\delta^P(\theta)\left(NP+\bar{P}^2-2P\bar{P}-\left(P-\bar{P}\right)^2\right) \qquad \text{(A.31)} \\
&= 2\bar{P}(\theta)\,\frac{N-\bar{P}(\theta)}{(N-1)\,N}-2\frac{1}{(N-1)\,N}\sum_{P=0}^{N}\delta^P(\theta)\left(P-\bar{P}\right)^2 \\
&= 2\left[\bar{P}(\theta)\,\frac{N-\bar{P}(\theta)}{(N-1)\,N}-\frac{\sigma_P^2(\theta)}{(N-1)\,N}\right]
\end{aligned}$$

That is, the dynamics of the expected value of $\widetilde{P}$ is given by:

$$\bar{P}(\theta+1)-\bar{P}(\theta)=2\alpha(1-\gamma)\left[\bar{P}(\theta)\frac{N-\bar{P}(\theta)}{(N-1)N}-\frac{\sigma_P^2(\theta)}{(N-1)N}\right] \tag{A.32}$$

or:

$$\bar{p}(\theta+1)-\bar{p}(\theta)=2\alpha(1-\gamma)\frac{1}{N-1}\left\{[1-\bar{p}(\theta)]\bar{p}(\theta)-\sigma_p^2(\theta)\right\} \tag{A.33}$$

Here, $\widetilde{p}(\theta)$ is the fraction of the population that has adopted the innovation after $\theta$ encounters, and $p$ is the realization of $\widetilde{p}$. If the number of encounters per unit of time is $\omega N$, we can make the substitution $t\omega N=\theta$ and express the dynamics in time units, where it is understood that the variables are now functions of the age of the innovation in time units:

$$\dot{\bar{p}}(t)=2\omega\alpha(1-\gamma)\frac{N}{N-1}\left\{[1-\bar{p}(t)]\bar{p}(t)-\sigma_p^2(t)\right\} \tag{A.34}$$

## B. THE DYNAMICS OF THE HETEROGENEITY

We want to separate terms that do not vanish in a steady state of the system from the rest. Differentiating 3.10 with respect to $t$, we find:

$$\begin{aligned}\dot{H} &= n(t)\bar{p}(0)\\ &+ \int_{-\infty}^{t} n(\omega)\,e^{Q(t-\omega,t)}\frac{dQ}{dt}\bar{p}(t-\omega)d\omega\\ &+ \int_{-\infty}^{t} n(\omega)\,e^{Q(t-\omega,t)}\bar{p}\,'(t-\omega)d\omega\end{aligned} \tag{B.1}$$

For the sake of simplicity, we denote the three terms of B.1 by $A$, $B$, and $C$.

$A$ does not need any further discussion.

In $B$ we add and subtract a term:

$$\rho\ln\left\{1-\widetilde{\lambda}[H(t)]\right\}\int_{-\infty}^{t} n(\omega)\,e^{Q(t-\omega,t)}\bar{p}(t-\omega)d\omega \tag{B.2}$$

$B$ can then be rewritten as:

$$\begin{aligned}B &= \int_{-\infty}^{t} n(\omega)\,e^{Q(t-\omega,t)}\left(\frac{dQ}{dt}-\rho\ln\left\{1-\widetilde{\lambda}[H(t)]\right\}\right)\bar{p}(t-\omega)d\omega\\ &+ \rho\ln\left\{1-\widetilde{\lambda}[H(t)]\right\}\int_{-\infty}^{t} n(\omega)\,e^{Q(t-\omega,t)}\bar{p}(t-\omega)d\omega\\ &= \int_{-\infty}^{t} n(\omega)\,e^{Q(t-\omega,t)}\left(\frac{dQ}{dt}-\rho\ln\left\{1-\widetilde{\lambda}[H(t)]\right\}\right)\bar{p}(t-\omega)d\omega\\ &+ \rho\ln\left\{1-\widetilde{\lambda}[H(t)]\right\}H(t)\end{aligned} \tag{B.3}$$

We evaluate the derivative $dQ/dt$:

$$\frac{dQ(t-\omega,t)}{dt} = Q_1 + Q_2 = \rho \ln\left\{1-\tilde{\lambda}\left[H(\omega)\right]\right\} + Q_2 \tag{B.4}$$

Here, $Q_1$ and $Q_2$ denote the partial derivatives with respect to the first and second arguments, respectively, of the function $Q$.

$B$ now becomes:

$$\begin{aligned} B &= \int_{-\infty}^{t} n(\omega)\, e^{Q(t-\omega,t)} \left(Q_2 + \rho \ln \frac{1-\tilde{\lambda}\left[H(\omega)\right]}{1-\tilde{\lambda}\left[H(t)\right]}\right) \bar{p}(t-\omega)\, d\omega \\ &+ \rho \ln\left\{1-\tilde{\lambda}\left[H(t)\right]\right\} H(t) \end{aligned} \tag{B.5}$$

$C$ can be integrated by parts:

$$\begin{aligned} C &= -n(\omega)\, e^{Q(t-\omega,t)}\, \bar{p}(t-\omega)\Big|_{-\infty}^{t} \\ &+ \int_{-\infty}^{t} n'(\omega)\, e^{Q(t-\omega,t)}\, \bar{p}(t-\omega)\, d\omega \\ &+ \int_{-\infty}^{t} n(\omega)\, e^{Q(t-\omega,t)} \frac{dQ}{d\omega}\, \bar{p}(t-\omega)\, d\omega \end{aligned} \tag{B.6}$$

We evaluate the derivative $dQ/d\omega$:

$$\frac{dQ}{d\omega} = -Q_1 = -\rho \ln\left\{1-\tilde{\lambda}\left[H(\omega)\right]\right\} \tag{B.7}$$

Evaluating the first term as well as adding and subtracting the term

$$n(t)\rho \int_{-\infty}^{t} e^{Q(t-\omega,t)} \ln\left\{1-\tilde{\lambda}\left[H(\omega)\right]\right\} \bar{p}(t-\omega)\, d\omega \tag{B.8}$$

we find the expression:

$$\begin{aligned} C &= -n(t)\,\bar{p}(0) \\ &+ \int_{-\infty}^{t} n'(\omega)\, e^{Q(t-\omega,t)}\, \bar{p}(t-\omega)\, d\omega \\ &+ \rho \int_{-\infty}^{t} \left[n(t)-n(\omega)\right] e^{Q(t-\omega,t)} \ln\left\{1-\tilde{\lambda}\left[H(\omega)\right]\right\} \bar{p}(t-\omega)\, d\omega \\ &- n(t)\rho \int_{-\infty}^{t} e^{Q(t-\omega,t)} \ln\left\{1-\tilde{\lambda}\left[H(\omega)\right]\right\} \bar{p}(t-\omega)\, d\omega \end{aligned} \tag{B.9}$$

Expression B.1 can now be written as:

$$\begin{aligned}\dot{H} &= \rho \ln\left\{1-\widetilde{\lambda}\left[H(t)\right]\right\} H(t) - n(t)\rho \int_{-\infty}^{t} \ln\left\{1-\widetilde{\lambda}\left[H(\omega)\right]\right\} e^{Q(t-\omega,t)}\, \bar{p}(t-\omega)d\omega \\ &+ \int_{-\infty}^{t} \left[ n(\omega)\left(Q_2 + \rho \ln \frac{1-\widetilde{\lambda}\left[H(\omega)\right]}{1-\widetilde{\lambda}\left[H(t)\right]}\right) + n'(\omega) \right. \\ &\left. + \rho\Big[n(t)-n(\omega)\Big] \ln\left\{1-\widetilde{\lambda}\left[H(\omega)\right]\right\} \right] e^{Q(t-\omega,t)}\, \bar{p}(t-\omega)d\omega \end{aligned} \tag{B.10}$$

Making the substitution $\tau = t - \omega$, we finally arrive at:

$$\begin{aligned}\dot{H} &= \rho \ln\left\{1-\widetilde{\lambda}\left[H(t)\right]\right\} H(t) - n(t)\rho \int_{0}^{\infty} \ln\left\{1-\widetilde{\lambda}\left[H(t-\tau)\right]\right\} e^{Q(\tau,t)}\, \bar{p}(\tau)d\tau \\ &+ \int_{0}^{\infty} \left[ n(t-\tau)\left(Q_2 + \rho \ln \frac{1-\widetilde{\lambda}\left[H(t-\tau)\right]}{1-\widetilde{\lambda}\left[H(t)\right]}\right) + n'(t-\tau) \right. \\ &\left. + \rho\Big[n(t)-n(t-\tau)\Big] \ln\left\{1-\widetilde{\lambda}\left[H(t-\tau)\right]\right\} \right] e^{Q(\tau,t)}\, \bar{p}(\tau)d\tau \end{aligned} \tag{B.11}$$

The second integral vanishes in a steady state. The properties of the steady states are hence determined by the first two terms.

## C. STEADY STATE

We take our point of departure in equation 3.12:

$$\ln\left[1-\widetilde{\lambda}(H)\right] H = n \int_{0}^{\infty} \ln\left[1-\widetilde{\lambda}(H)\right] e^{\widetilde{Q}(\tau,H)}\, \bar{p}(\tau)d\tau \tag{C.1}$$

Using the fact that in the steady state

$$\begin{aligned} e^{\widetilde{Q}(\tau,t)} &= \exp\left[\rho \int_{0}^{\tau} \ln\left[1-\widetilde{\lambda}(H)\right] d\kappa\right] \\ &= \exp\left[\rho \ln\left[1-\widetilde{\lambda}(H)\right] \int_{0}^{\tau} d\kappa\right] \\ &= \exp\left[\ln\left[1-\widetilde{\lambda}(H)\right]^{\rho\tau}\right] \\ &= \left[1-\widetilde{\lambda}(H)\right]^{\rho\tau} \end{aligned} \tag{C.2}$$

this equation, for $\widetilde{\lambda}(H) > 0$, becomes:

$$H = n \int_{0}^{\infty} \left[1-\widetilde{\lambda}(H)\right]^{\rho\tau} \bar{p}(\tau)d\tau \tag{C.3}$$

We define $\psi(\tau; H, \rho) > 0$ by:

$$\begin{aligned}
\psi(\tau; H, \rho) &:= \frac{\left[1 - \widetilde{\lambda}(H)\right]^{\rho\tau}}{\int_0^\infty \left[1 - \widetilde{\lambda}(H)\right]^{\rho\kappa} d\kappa} \\
&= \frac{\left[1 - \widetilde{\lambda}(H)\right]^{\rho\tau}}{\left[1 - \widetilde{\lambda}(H)\right]^{\rho\kappa}\Big|_0^\infty} \rho \ln\left[1 - \widetilde{\lambda}(H)\right] \qquad \text{(C.4)} \\
&= -\rho \ln\left[1 - \widetilde{\lambda}(H)\right]\left[1 - \widetilde{\lambda}(H)\right]^{\rho\tau}
\end{aligned}$$

Clearly, $\psi(\tau; H, \rho)$ integrates to one:

$$\int_0^\infty \psi(\tau; H, \rho) d\tau = 1 \qquad \text{(C.5)}$$

Multiplying both sides of equation C.3 by $-\rho \ln\left[1 - \widetilde{\lambda}(H)\right]$, we can rewrite it as:

$$-\rho \ln\left[1 - \widetilde{\lambda}(H)\right] H = n \int_0^\infty \psi(\tau; H, \rho)\, \bar{p}(\tau) d\tau \qquad \text{(C.6)}$$

The integral multiplying $n$ is, hence, a weighted average of $\bar{p}$ over the age of the innovation with the weights decreasing with increasing age. Since $0 < \bar{p} < 1$, the weighted average also lies between 0 and 1. If $\widetilde{\lambda}$ increases, more weight is given to small values of $\tau$. Since $\bar{p}(\tau)$ increases with $\tau$, the value of the integral decreases with larger values of $\widetilde{\lambda}$. The rate of absorption $\widetilde{\lambda}$, however, decreases with an increase in $H$. Hence the value of the integral increases with an increase in $H$. We write the integral as $1 > \imath(H, \rho) > 0$, $\partial\imath/\partial H \geq 0$.

Equation C.3 now becomes:

$$-\rho \ln\left[1 - \widetilde{\lambda}(H)\right] H = n\, \imath(H, \rho) \qquad \text{(C.7)}$$

or:

$$n = -\rho \ln\left[1 - \widetilde{\lambda}(H)\right] \frac{H}{\imath(H, \rho)} =: g(H, \rho) \qquad \text{(C.8)}$$

The function that we have defined as $g(H, \rho)$ describes the long-run capacity of the language to absorb innovations without an increase in the heterogeneity. The form of $g$ depends on how $\widetilde{\lambda}$ behaves for large values of $H$. It is clear that $g(0, \rho) = 0$ if $\widetilde{\lambda}(0) > 0$. Also, if for some value $H^M$ of $H$ the value of the function $\widetilde{\lambda}(H^M) = 0$, then $g(H^M, \rho) = 0$. If $\widetilde{\lambda}(H) \to 0$ as $H \to \infty$, the behavior of $g$ depends on "how fast" $\widetilde{\lambda}(H)$ approaches zero. It has to be faster than $1/H$ for $g$ to approach zero for large values of $H$.

We will assume that either there exists an $H^M$ such that $\widetilde{\lambda}(H^M) = 0$ or that $\widetilde{\lambda}(H)$ approaches zero "fast enough" for sufficiently high $H$. Then, the function $g$ has the general form of figure 4.

# Comparing Three Plagiarism Tools (Ferret, Sherlock, and Turnitin)

**Mitra Shahabi**[1] *mitra.shahabi@ua.pt*
*Department of Language and Culture*
*University of Aveiro*
*Aveiro, 3800-356, Portugal*

**Abstract**

An attempt was made to carry out an experiment with three plagiarism detection tools (two free/open source tools, namely, Ferret and Sherlock, and one commercial web-based software called Turnitin) on Clough-Stevenson's corpus [1] including documents classified in three types of plagiarism and one type of non-plagiarism. The experiment was toward Extrinsic/External detecting plagiarism. The goal was to observe the performance of the tools on the corpus and then to analyze, compare, and discuss the outputs and, finally to see whether the tools' identification of documents is the same as that identified by Clough and Stevenson.

**Keywords:** Plagiarism detection tool, Ferret, Sherlock, Turnitin, Clough-Stevenson's corpus.

## 1. INTRODUCTION

Plagiarism, defined as the act of using others´ ideas and words in a text document without acknowledging the sources, is one of the most increasing issues in academic communities especially for the higher education institutions [2]. The existence of Internet and online search engines has advanced the international collaboration in education but at the same time it also has raised the plagiarism opportunity. Nowadays, pre-written essays are accessible online through the websites, essay banks or paper mills. This technology can be misused by the students and lead them to plagiarism.

Motivated by the plagiarism problem, a field namely plagiarism detection arises. Both the academic and commercial communities put their effort to detect plagiarism [1]. Plagiarism analysis can be distinguished as intrinsic and extrinsic analysis [3]. In intrinsic analysis, the aim is to detect plagiarism within the document (i.e. the source does not to be identified); whilst in extrinsic analysis, the aim is to detect plagiarism across documents (i.e. comparing suspicious documents with their potential sources).

Plagiarism detection methods in natural language originate from diverse areas such as file comparison, information retrieval, authorship attribution, file compression, and copy detection. These approaches work well to handle text with minimal alterations such as word-for-word plagiarism. However, they still have problems in detecting paraphrasing plagiarism, plagiarism of ideas, and cross-lingual plagiarism where the text is altered significantly [1]. The academic and commercial communities are still in the process of delivering a better plagiarism detection solution; see for example the three competitions on plagiarism detection in the recent years: PAN´09, PAN´10, and PAN´11. PAN´11 was held in conjunction with 2011 CLEF conference [4]; eleven plagiarism detection were evaluated based on the third revised edition of the PAN plagiarism corpus PAN-PC-11. Figure 1 shows the overview of important corpus parameters [4].

Comparing the detection performance measures of plagdet, precision, recall, and granularity of

[1] PhD student in Translation with scholarship from Fundação para a Ciência e a Tecnologia (FCT) (Portugal), with reference number SFRH/BD/60210/2009

the detectors[2], Grman and Raven [5] was known as the best-performing detector and Grozea and Popescu [6] and Oberreuter et al. [7] were known as the second and the third best-performing tools, respectively (cited in [4]).

**Document Statistics**

| Document Purpose | | Plagiarism per Document | | Document Length | |
|---|---|---|---|---|---|
| source documents | 50% | hardly (5%-20%) | 57% | short (1-10 pp.) | 50% |
| suspicious documents | | medium (20%-50%) | 15% | medium (10-100 pp.) | 35% |
| – with plagiarism | 25% | much (50%-80%) | 18% | long (100-1000 pp.) | 15% |
| – without plagiarism | 25% | entirely (>80%) | 10% | | |

**Plagiarism Case Statistics**

| Obfuscation | | Case Length | |
|---|---|---|---|
| none | 18% | short (<150 words) | 35% |
| paraphrasing | | medium (150-1150 words) | 38% |
| – automatic (low) | 32% | long (>1150 words) | 27% |
| – automatic (high) | 31% | | |
| – manual | 8% | | |
| translation ({de, es} to en) | | | |
| – automatic | 10% | | |
| – automatic + manual correction | 1% | | |

**FIGURE 1:** A screenshot of the corpus statistics for 26 939 documents and 61 064 plagiarism cases in the PAN-PC-11.

In comparison with the performance reported in PAN´09 and PAN´10, a PAN´11 shows a drop in the plagdet performance; this result has been attributed to an increased detection difficulty [4].

There are different plagiarism detection tools among which we can refer to Turnitin, Glatt, Eve2, Wordcheck, CopyCatchGold, and so on [8; 9; 10; 11; 12; 13].

The tools performance is usually based on two methods, statistical, semantical, or both. However, the statistical method are better welcomed since they are easily applicable

In this study, an extrinsic plagiarism detection experiment was conducted. The applied detection tools were using three tools Ferret [14], Sherlock [15] and Turnitin, which is an online service created by iParadigms, LLC. The rest of this document will explain the details of the tools and corpus, discussion of the experiment results, and conclusion of the experiment.

## 2. THE COURPUS

In this study, the freely available Clough-Stevenson's corpus [1] was applied. The corpus consists of answers to five short questions on a variety of topics in Computer Science field. The five short questions are:

1. What is inheritance in object oriented programming?

[2]

$$prec(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\bigcup_{s\in S}(\mathbf{s}\sqcap\mathbf{r})|}{|\mathbf{r}|},\ rec(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\bigcup_{r\in R}(\mathbf{s}\sqcap\mathbf{r})|}{|\mathbf{s}|},$$

$$gran(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_s|,\quad plagdet(S,R) = \frac{F_1}{\log_2(1+gran(S,R))}$$

S: the set of plagiarism in the corpus; R: the set of detection reported by a plagiarism detector for the suspicious document; F1: the equally weighted harmonic mean of precision and recall. Plagdet is the combination of the other three measures.

2. Explain the PageRank algorithm that is used by the Google search engine.
3. Explain the Vector Space Model that is used for Information Retrieval.
4. Explain Bayes Theorem from probability theory.
5. What is dynamic programming?

To simulate plagiarism, for each question, a suitable entry in Wikipedia which contains the answer to the question was selected as the source document. In order to represent a variety of different degrees of plagiarism, participants were asked to answer the question using one of the following models (pp. 7-8):

*Near Copy*: Participants were asked to answer the questions by performing copy-and-paste action from the relevant Wikipedia entry of 200-300 words without any instruction about which parts of the article to copy.

*Light Revision*: Participants were asked to answer the questions by performing copy-and-paste action from the relevant Wikipedia entry and they may alter it in some basic ways such as substituting words and phrases with synonyms and also paraphrasing. However, they are not allowed to alter the order of information found in the sentences.

*Heavy Revision*: Participants were asked to answer the questions by performing copy-and-paste action from the relevant Wikipedia entry and instructed to rephrase the text without any constraint about how to alter the text.

*Non-plagiarism*: Learning materials such as lecture notes or textbooks sections that are relevant with the questions were provided to the participants. They were asked to answer the questions by using their own knowledge including what they had learned from the materials provided. Participants were allowed to look at other materials but Wikipedia to answer the questions.

Accordingly, the corpus consists of 100 documents: five Wikipedia entries and 95 answers provided by 19 participants. A breakdown of the number of answers in the corpus can be seen in Table 1. The average length of file in the corpus is 208 words and 113 tokens. 59 of the files are written by native English speakers and the remaining 36 files by non-native speakers.

| Category | Learning Task | | | | | Total |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | |
| Near Copy | 4 | 3 | 3 | 4 | 5 | 19 |
| Light Revisions | 3 | 3 | 4 | 5 | 4 | 19 |
| Heavy Revisions | 3 | 4 | 5 | 4 | 3 | 19 |
| Non-plagiarism | 9 | 9 | 7 | 6 | 7 | 38 |
| Total | 19 | 19 | 19 | 19 | 19 | 95 |

**TABLE 1:** Corpus breakdown

## 3. THE PLAGIRISM DETECTION TOOLS

Plagiarism detection tools are useful in terms of detecting and also preventing plagiarism. Since there are many tools available now, one should be wise on selecting it according to their need. And also, as plagiarism detection software only gives suggestion to the user about the suspicious documents, further analysis should be done by human as well as the final decision.

For this study, the three plagiarism detection tools Ferret, Sherlock, and Turnitin were compared and analyzed. The systems detect plagiarism based on the statistical methods of matching n-gram words (adjacent `words´ of input), between the texts. The comparison is carried out between all the documents, i.e. every document is compared with every other document. As the

tools read the documents they extract all n-grams of the two documents under the comparison and then match them. Afterwards, they calculate the rate of documents similarity based on the following formula, where A is "the set of n-grams extracted from one of the documents and B is the set of n-grams from the comparing document by [16].

$$Similarity = \frac{Number\ of\ common\ trigrams}{Total\ number\ of\ trigrams} = \frac{A \cap B}{A \cup B}$$

### 3.1. Ferret

Ferret is a freely available standalone plagiarism detection system developed at the University of Hertfordshire. It runs on Windows environment and very easy to install and run. File formats that Ferret can process are .txt, .rtf, .doc and .pdf. The algorithm is written in C++. Ferret takes a set of documents, converts each text into reference number, set of characteristic trigrams. It compares every text with each other based on counting the number of distinct trigrams similar between the texts, and produces a list of file-pairs together with the similarity scores that ranked from the most similar pair to the least similar one. This count is used to calculate the resemblance measure, as the number of similar trigrams in a pair of documents, divided by the total number of different trigrams in the pair. Ferret manifests the scores of similarity precisely like 0.90991. The numbers were rounded for sake of being simplified for analysis; in this case, for example, it was taken as 0.91. The system allows user to select any pair of texts and do further investigation as they will be displayed side by side with similar paragraphs highlighted (similar parts in blue and different parts in black). See the Figures 2 and 3.

### 3.2. Sherlock

Sherlock is a free and open source plagiarism detection program for essays, computer source code files, and other kinds of textual documents in digital form. It turns the texts into digital signatures to measure the similarity between the documents. A digital signature is a number formed by turning several words (3 by default) in the input into a series of bits and joining those bits into a number.

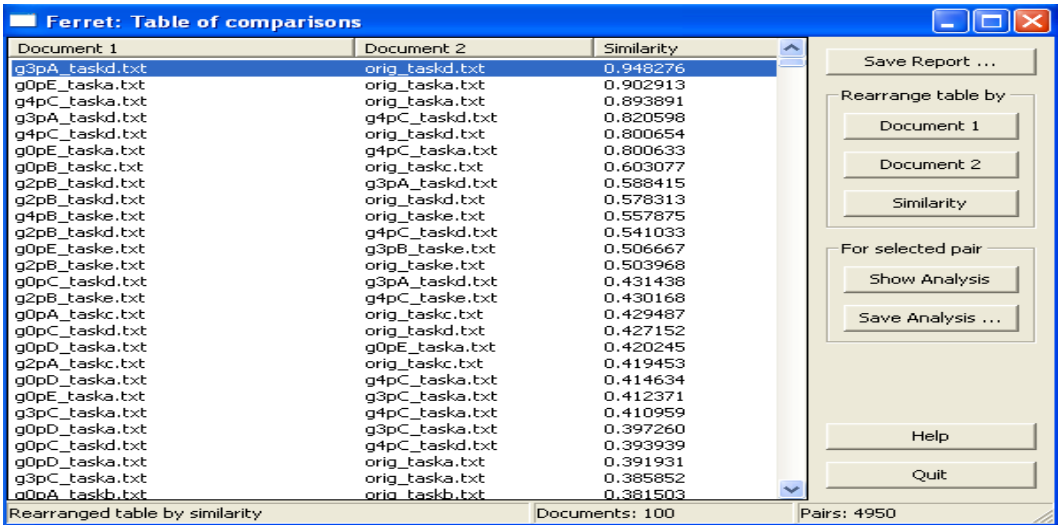



**FIGURE 2:** A screenshot of Ferret showing a table of comparison

Sherlock is written in C programming language (Fig. 4) and needs to be compiled before being installed either on Unix/Linux or Windows. It is a command-line program and it does not have a graphical user interface. Executing a "sherlock *.txt" command will compare all the text files in the current directory and produce a list of file-pairs together with the similarity percentage (Fig. 5). This output list is not ordered by the similarity percentage.

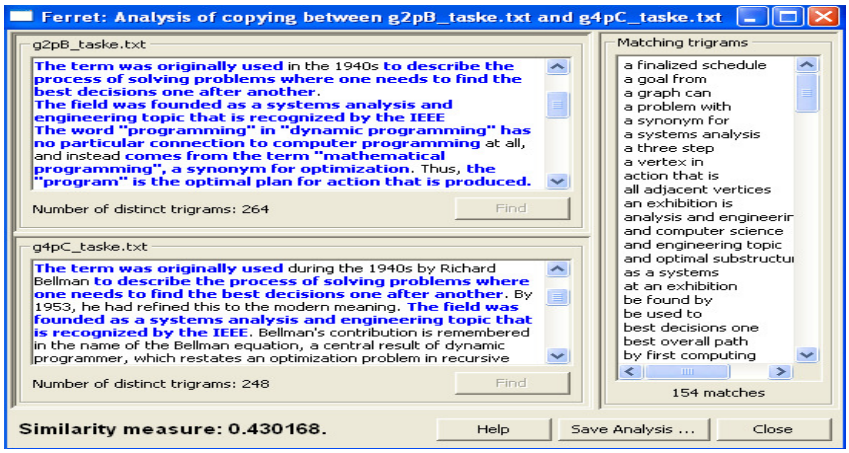



**FIGURE 3:** A screenshot of Ferret showing the analysis of copying between two texts

Important point to be noted when analyzing the output of Sherlock is the fact that 100% score does not imply that the files are identical because the Sherlock program actually throws away some data randomly in the process in order to simplify and speed up the match.

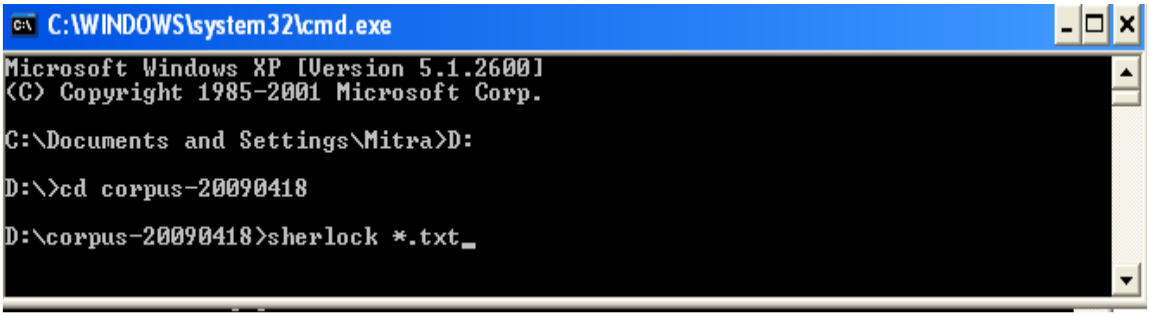



**FIGURE 4:** A screenshot of Sherlock showing a command-line

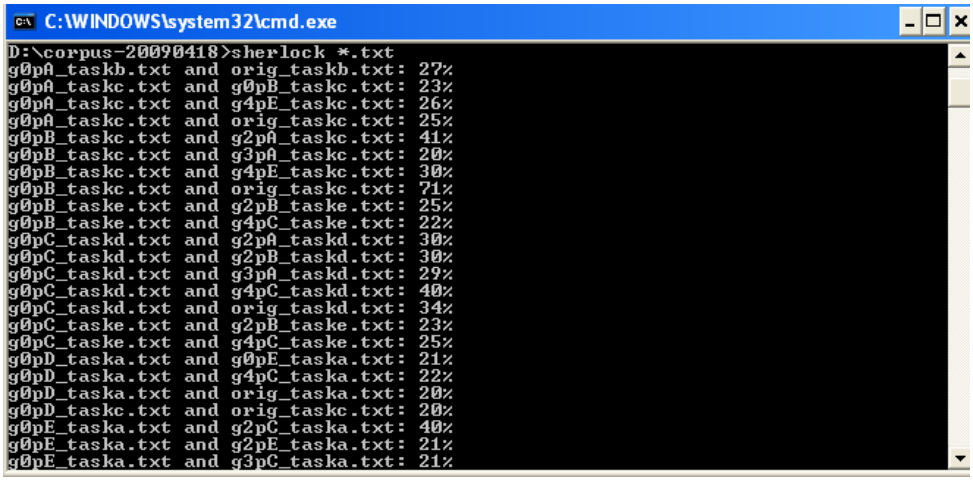



**FIGURE 5:** A screenshot of Sherlock showing the results the similarity of the compared documents

There are four command-line options giving a possibility to change the numbers in the command line in order to see different performance results.

a) *-t threshold%.* The system is showing the files with similarities of 20% by default; the higher this threshold the more similar files are printed.
b) *-z zerobits*. The 'granularity' of the comparison is 4 by default but it can be changed from 0 to 31. However, it should be noticed that the higher this number, the less exact the comparison will be but the faster, and vice versa.
c) *-n number_of_words*. The default for the system is 3 words (3-gram) form one digital signature. We can change the number of words (min 1, max 7); the higher the number the slower but more exact the process however "the less likely they are to co-occur in both texts" (Specia, 2010), and vice versa.
d) *-o outfile*. It is to store the different results, acquired by making some changes in the aforementioned program options, in the same folder that the corpus exists.

Example: sherlock -t 80% -z 3 -n 2 -o results.txt *.java (see Fig. 4).

With Sherlock, it is not possible to see what parts of the compared documents are similar. It is only possible to see the rate of similarity of the documents in question in percentage (see Fig. 5).

### 3.3. Turnitin

Turnitin is a web-based subscription plagiarism detection service, maintained by a company named iParadigms. To use this service, user simply has to log on to Turnitin website without any other installation. Turnitin detects material copied from the Internet and also cross-checking of submitted essays within a task as well as other text documents in the database. Every submitted essay is added to the database and will be used in the future when other essay is submitted. Turnitin offers a free restricted trial account that allows user to submit five text documents over 30 days period. In this trial account, access to the Turnitin database is not given.

In Turnitin, we cannot have, like Ferret, both the documents in one window to see the similarities of the compared texts. The only document that is shown is the suspicious text; the parts similar to the other document appear in red the distinct parts are in black color (see Fig. 6).


**FIGURE 6:** A screenshot of Turnitin showing the results the similarity of the compared documents

## 4. METHODOLOY

The experiment was carried out with the three tools on the corpus. The present study did not cover all the results reported by the three tools; the focus was only on the results of comparison between the students' documents (tasks a to e) and their related original sources (original a to e). The results of comparison between the student' documents, or in case of Turnitin the comparison

with other sources, were left.

For Sherlock, *t* (threshold) was changed from 0.20 to 0.00 in order to make the tool compatible with Ferret and Turnitin which report the similarities from 0.00.

After analyzing the differences and similarities between the three tools, the goal was to find whether or not their outputs match the classification of the tasks presented by Clough and Stevenson.

As the outputs of all systems appeared in numbers, the Clough-Stevenson's classification of documents (Appendix B) was also needed in numbers; hence, the mean similarity between the documents and the Wikipedia articles illustrated by Clough and Stevenson [1] (p.14) was used for this purpose. See Figure 7.

| Category | $c_n(A,B)$ for $n$-gram | | | | | $lcs_{norm}$ |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| Near copy | 0.95 | 0.89 | 0.85 | 0.81 | 0.78 | 0.88 |
| Light revision | 0.87 | 0.70 | 0.56 | 0.46 | 0.39 | 0.76 |
| Heavy revision | 0.81 | 0.52 | 0.34 | 0.26 | 0.21 | 0.58 |
| Non-plagiarised | 0.63 | 0.23 | 0.05 | 0.01 | 0.00 | 0.41 |

**FIGURE 7:** mean similarity between the documents and the Wikipedia articles illustrated by Clough and Stevenson

## 5. ANALYSIS AND DISCUSSION

Appendix A shows the results of all the three systems along with Clough-Stevenson's classification of the documents Ferret and Sherlock, in most cases, reported the results more or less the same, but Turnitin's outputs in many cases were greatly different from the other two, usually showing a higher percentage of similarities (Appendix A). In order to investigate the reason, The system´s 'analysis part' was checked to see the overlapped parts of the two documents in order to examine whether or not the tools have matched the compared documents properly. It could be realized only with Ferret and Turnitin, because as aforementioned before Sherlock has the drawback of not providing a graphical user interface showing the two documents with the overlapped and distinct parts; it just reports the percentages results.

It was discovered that Turnitin performs quite well and it is Ferret that does not show the expected percentage, because it considers the longer text (for this corpus, the longer is always the source [1]) as the base and then looks how much of this text is overlapped by the shorter text and the result is shown as the percentage of similarity between the two documents[3], i.e. if the suspicious document is, for example, 100% similar to the original document but its size covers only 40% of the original source, Instead of reporting 100% plagiarism, Ferret reports 40% plagiarism.

Regarding the fact that Ferret and Sherlock reported a quite similar output it was speculated that Sherlock, probably, performs like Ferret. And because of the problems addressed to Ferret and Sherlock, the comparison was only made between the Turnitin's output and Clough-Stevenson's classification.

Analyzing the data in Appendix A, it was discovered that out of 95 documents, Turnitin identified 61 documents similar to and 34 documents different from Clough-Stevenson's classification of documents. Table 2 illustrates these 34 cases. The system acted properly for all the non-plagiarized tasks; the outputs match with Clough-Stevenson's. The differences up to 0.20 between Turnitin outputs and Clough-Stevenson's classification of the documents was ignored since, for Clough-Stevenson's classification of the texts, the mean similarity was considered for

[3] It is infact the shorter text which must be checked how much of it has been covered by the original text.

comparison; however, for Turnitin's the exact percentage of similarity was taken into account.

Table 2 shows the documents whose rate of plagiarism has been wrongly reported by Turnitin. The figures in blue indicate 0.40 ≤ 0.20 differences between the results of the system and the Clough and Stevenson´s; and the reds signify a considerable difference (≥ 0.40) between them.

| | Document 1 | Document 2 | Clough -Stevenson (mean similarity) | Turnitin |
|---|---|---|---|---|
| 3 | g0pA_taskc.txt | orig_taskc.txt | 0.56 | 0.85 |
| 4 | g0pA_taskd.txt | orig_taskd.txt | 0.34 | 0.00 |
| 11 | g0pC_taska.txt | orig_taska.txt | 0.34 | 0.00 |
| 15 | g0pC_taske.txt | orig_taske.txt | 0.56 | 0.89 |
| 17 | g0pD_taskb.txt | orig_taskb.txt | 0.56 | 0.76* |
| 18 | g0pD_taskc.txt | orig_taskc.txt | 0.34 | 0.58 |
| 21 | g0pE_taska.txt | orig_taska.txt | 0.56 | 0.99 |
| 22 | g0pE_taskb.txt | orig_taskb.txt | 0.34 | 0.66 |
| 27 | g1pA_taskb.txt | orig_taskb.txt | 0.34 | 0.00 |
| 28 | g1pA_taskc.txt | orig_taskc.txt | 0.56 | 0.26 |
| 29 | g1pA_taskd.txt | orig_taskd.txt | 0.85 | 0.34 |
| 34 | g1pB_taskd.txt | orig_taskd.txt | 0.56 | 0.35 |
| 35 | g1pB_taske.txt | orig_taske.txt | 0.85 | 0.50 |
| 36 | g1pD_taska.txt | orig_taska.txt | 0.56 | 0.34 |
| 42 | g2pA_taskb.txt | orig_taskb.txt | 0.34 | 0.00 |
| 43 | g2pA_taskc.txt | orig_taskc.txt | 0.56 | 0.78* |
| 44 | g2pA_taskd.txt | orig_taskd.txt | 0.85 | 0.31 |
| 48 | g2pB_taskc.txt | orig_taskc.txt | 0.34 | 0.00 |
| 49 | g2pB_taskd.txt | orig_taskd.txt | 0.56 | 0.93 |
| 51 | g2pC_taska.txt | orig_taska.txt | 0.85 | 0.66* |
| 54 | g2pC_taskd.txt | orig_taskd.txt | 0.34 | 0.56 |
| 55 | g2pC_taske.txt | orig_taske.txt | 0.56 | 0.00 |
| 62 | g3pA_taskb.txt | orig_taskb.txt | 0.34 | 0.00 |
| 68 | g3pB_taskc.txt | orig_taskc.txt | 0.34 | 0.00 |
| 69 | g3pB_taskd.txt | orig_taskd.txt | 0.56 | 0.30 |
| 75 | g3pC_taske.txt | orig_taske.txt | 0.56 | 0.78* |
| 78 | g4pB_taskc.txt | orig_taskc.txt | 0.34 | 0.61 |
| 79 | g4pB_taskd.txt | orig_taskd.txt | 0.56 | 0.82 |
| 84 | g4pC_taskd.txt | orig_taskd.txt | 0.34 | 0.97 |
| 85 | g4pC_taske.txt | orig_taske.txt | 0.56 | 0.91 |
| 86 | g4pD_taska.txt | orig_taska.txt | 0.56 | 0.28 |
| 91 | g4pE_taska.txt | orig_taska.txt | 0.34 | 0.00 |
| 92 | g4pE_taskb.txt | orig_taskb.txt | 0.56 | 0.93 |
| 93 | g4pE_taskc.txt | orig_taskc.txt | 0.85 | 0.32 |

**TABLE 2:** The differences between Turnitin's output and Clough-Stevenson's classifications[4].

In order to simplify the results, the wrong outputs are presented below in table 3, in the way

[4] As this table has been, in fact, extracted from the table in Appendix A, the numbers in the left column seem out of order.

Clough and Stevenson classified the texts (near-copy, heavy revision, light revision, and non-plagiarism). If the differences between the Turnitin´s figures are higher than 20 it is an indication of changing the level of the text in the classification of texts; that is, the text with 0.56 rate of plagiarism, being classified by Clough and Stevenson as highly revised, was seen in turnitin´s outputs with 0.35 rate of plagiarism; so it was reported as a lightly revised text in turnitin´s results. It can be concluded that the blue colors mean texts with one level higher or lower than the real classification of the text, and the red color identifies two levels higher or lower than the accurate position; except for light revision and near-copy texts which the difference of the rate of their plagiarism is ≤0.30); they have been marked with (*). The differences are summarized in table 3 below.

| | Turnitin´s | | | |
|---|---|---|---|---|
| Clough & Stevenson´s | Near-copy | High revision | Light revision | Non-plagiarism |
| Near-copy | – | – | – | – |
| High revision | 8 | – | 4 | 1 |
| Light revision | 1 | 5 | – | 7 |
| Non-plagiarism | – | 3 | 1 | – |

**TABLE 3:** Differences of texts classification between Turnitin´s outputs and Clough and Stevenson´s classification of texts

As noticed in table 3, the noises produced by Turnitin are as follows: 8 highly revised texts were reported near-copy, 4 were reported lightly revised, and 1 as non-plagiarized; one lightly revised texts was reported near-copy, 5 were reported as highly revised, and 7 as non-plagiarized; and 3 non-plagiarized texts were reported as highly revised and 1 as lightly revised.

Regarding the fact that Ferret and Sherlock reported a quite similar output it was speculated that Sherlock, probably, performs like Ferret. Although they did not report the results in a manner expected (like Turnitin), their outputs were evaluated in terms of precision and recall (Table 4). Only file-pairs of answer and source within the same task were included for the evaluation.

As noticed in table 4, both Ferret and Sherlock give a perfect precision score for all cases, starting from similarity score 0.1 for Ferret and similarity percentage 10% for Sherlock, which means all captured documents are indeed plagiarism. However, both systems give a very low recall score when thresholds are set very high (0.5 for Ferret and 50% for Sherlock). As the thresholds are set lower, the recall scores are getting higher. At similarity score threshold 0.1, recall score of Ferret is 0.68421053 where 39 out of 57 cases of plagiarism detected. At similarity percentage threshold 10%, recall score of Sherlock is 0.57894737 where 33 out of 57 cases of plagiarism detected.

| Ferret | | | Sherlock | | |
|---|---|---|---|---|---|
| | Precision | Recall | | Precision | Recall |
| ≥ 0.5 | 1 | 0.14035088 | ≥ 50% | 1 | 0.10526316 |
| ≥ 0.4 | 1 | 0.19298246 | ≥ 40% | 1 | 0.1754386 |
| ≥ 0.3 | 1 | 0.31578947 | ≥ 30% | 1 | 0.22807018 |
| ≥ 0.2 | 1 | 0.45614035 | ≥ 20% | 1 | 0.36842105 |
| ≥ 0.1 | 1 | 0.68421053 | ≥ 10% | 1 | 0.57894737 |

**TABLE 4**: Precision and Recall of Ferret and Sherlock Outputs

In Ferret output, the majorities of the file-pairs captured with similarity score ≥ 0.2 are near copy and light revision plagiarism. Among 26 suspicious documents, only four of them are categorized as heavy revision and all these four texts are written by non-native English speakers. The file-pairs captured with similarity score 0.1-0.2 vary between near copy, light revision, and heavy

revision plagiarism. All of the non-plagiarism answers have similarity score below 0.03. There are also three heavy revision plagiarism texts within this range as well as one near copy plagiarism text written by a non-native English speaker but he/she claims a very good knowledge of the question topic and the question is perceived as a not difficult one. There is only one document that contains plagiarism but has zero similarity score against its original.

Similar to Ferret, in Sherlock output, the majorities of the file-pairs captured with similarity percentage ≥ 30% are near copy and light revision plagiarism. Only one of 13 suspicious documents is a heavy revision and it is written by a non-native English speaker. The rest of the heavy plagiarism answers have similarity percentage below 30% along with the other near copy and light revision plagiarism. There are 15 texts which have similarity percentage between 1%-9% and three of them are non-plagiarism. Setting the threshold to 1% will give 45 (out of 57) documents that contains plagiarism with different degrees and three non-plagiarism documents, which implies there are 12 documents that actually contains plagiarism but assigned a similarity percentage of zero.

## 6. CONCLUSION

In this paper it was tried to reveal some strengths and weaknesses of three plagiarism detection tools, namely, Sherlock, Ferret, and Turnitin. They were compared according to their features and performances. The criterion for selecting these tools for this study was to discover how the easily available or free/open source tools are performing and at the end which of them can be considered the best. Since one of the advantages of open source tools is that we can improve them in order to meet our goals. It appeared that Ferret and Sherlock, in most cases, produce the same results in plagiarism detection performance; however, Turnitin reported the results with great difference from the other two tools: It showed a higher percentage of similarities between the documents and the source. After investigating the reason (just checked with Ferret and Turnitin, cause Sherlock does not provide a view of the two documents with the overlapped and distinct parts), it was discovered that Turnitin performs quite acceptable and it is Ferret that does not show the expected percentage; it considers the longer text (for this corpus the longer is always the source) as the base and then looks how much of this text is overlapped by the shorter text and the result is shown as the percentage of similarity between the two documents, and this leads to wrong results. Therefore, there is always a need for human intervention to make a lot of effort to check if the output reports a real percentage of plagiarism. From this it can be also speculated that Sherlock does not manifest the results properly. Although they did not report the results in a manner expected (like Turnitin), their outputs were evaluated in terms of precision and recall.

Both Ferret and Sherlock give a perfect precision score for all cases, which means all captured documents are indeed plagiarism. However, both systems give a very low recall score when thresholds are set very high. As the thresholds are set lower, the recall scores are getting higher.

## 7. Future work

A change in Ferret system program can probably solve the problem of giving wrong percentage, because its problem seems just in giving the non-intended percentage, and it works well in matching the 3-grams. One negative point of Sherlock is the user interface; It does not have a graphical user interface, i.e., it does not manifest the content of the texts in condition we need to analyze the output. It is very important that a user be able to easily compare the parts that are marked similar. For this purpose it is better that the tool displays the comparing files next to each other with highlighting similar parts

The reliability of the Clough-Stevenson's corpus, as the only base of evaluation, is also questionable.

## 8. REFERENCES


1. P. Clough and M. Stevenson. “Developing a Corpus of Plagiarized Short Answers, Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis, In Press.” Internet: http://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html#Download, Sep. 10, 2009 [Oct. 12, 2011].

2. G. Judge. “Plagiarism: Bringing Economics and Educations Together (With a Little Help from IT).” Computers in Higher Economic Review, vol. 20(1), pp. 21-26, 2008.

3. B. Stein and S. Meyer zu Eissen. “Near similarity search and plagiarism analysis,” in From Data and Information Analysis to Knowledge Engineering, M. Spiliopoulou et al., EDs. Springer, 2006, pp. 430-437.

4. M. Potthast et al. “Overview of the 3rd international competition in plagiarism detection”: notebook for PAN at CLEF 2011, in Notebook Papers of CLEF 2011 LABs and Workshops, 19-22 Sep., Amsterdam, The Netherlands, 2011.

5. J. Grman and R. Ravas. “Improved implementation for finding text similarities in large collection of data: notebook for PAN at CLEF 2011, in Notebook Papers of CLEF 2011 LABs and Workshops, 19-22 Sep., Amsterdam, The Netherlands, 2011.

6. C. Grozea and M. Poescu. “The encoplot similarity measure for automatic detection of plagiarism”: notebook for PAN at CLEF 2011, in Notebook Papers of CLEF 2011 LABs and Workshops, 19-22 Sep., Amsterdam, The Netherlands, 2011.

7. G. Oberreuter, G. L´Huillier, S. Apíos, and J. D. Velasquez. “Approaches for intrinsic and external plagiarism detection”: notebook for PAN at CLEF 2011, in Notebook Papers of CLEF 2011 LABs and Workshops, 19-22 Sep., Amsterdam, The Netherlands, 2011.

8. M. Delvin. “Plagiarism detection software: how effective is it? Assessing Learning in Australian Universities.” Internet: http://www.cshe.unimelb.edu.au/assessing learning/docs/PlagSoftware.pdf, 2002 [Sep. 23, 2012].

9. T. Lancaster and F. Culwin. “A review of electronic services for plagiarism detection in student submissions.” the Teaching of Computing, Edinburgh, 2000. Internet: http://www.ics.heacademy.ac.uk/events/presentations/317_Culwin.pdf, 2000 [Oct. 01, 2012].

10. T. Lancaster and F. Culwin. “Classifications of Plagiarism Detection Engines.” *ITALICS*, vol. 4 (2), 2005.

11. H. Maurer, F. Kappe, and B. Zaka. “Plagiarism – A Survey.” *Journal of Universal Computer Sciences*, vol. 12 (8), pp. 1050 – 1084, 2006.

12. C. J. Neill and G. Shanmuganthan. “A Web – enabled plagiarism detection tool.” *IT Professional*, vol. 6 (5), pp. 19 – 23, 2004.

13. C. Lyon, R. Barrett and J. Malcolm. “A theoretical basis to the automated detection of copying between texts and its practical implementation in the Ferret plagiarism and collusion detector,” in Proc. The Plagiarism: Prevention, Practice and Policies Conference, 2004.

14. R. Pike. “The Sherlock Plagiarism Detector.” Internet: http://www.cs.su.oz.au/~scilect/sherlock, 2007 [Oct. 04, 2011].

15. J. Malcolm and P. Lane. “Efficient Search for Plagiarism on the Web.” *Kuwait*, vol. 1, pp. 206-211, 2008.

# APPENDICES

**APPENDIX A:** The results shown by the three systems & Clough and Stevenson´s mean similarity of documents

| | Documents | | Plagiarism detection tools | | | |
|---|---|---|---|---|---|---|
| | Document 1 | Document 2 | Clough-Stevenson (mean similarity) | Ferret | Sherlock | Turnitin |
| 1 | g0pA_taska.txt | orig_taska.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 2 | g0pA_taskb.txt | orig_taskb.txt | 0.85 | 0.38 | 0.27 | 1.00 |
| 3 | g0pA_taskc.txt | orig_taskc.txt | 0.56 | 0.42 | 0.25 | 0.85 |
| 4 | g0pA_taskd.txt | orig_taskd.txt | 0.34 | 0.06 | 0.00 | 0.00 |
| 5 | g0pA_taske.txt | orig_taske.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 6 | g0pB_taska.txt | orig_taska.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 7 | g0pB_taskb.txt | orig_taskb.txt | 0.05 | 0.01 | 0.00 | 0.00 |
| 8 | g0pB_taskc.txt | orig_taskc.txt | 0.85 | 0.60 | 0.71 | 0.74 |
| 9 | g0pB_taskd.txt | orig_taskd.txt | 0.56 | 0.22 | 0.16 | 0.58 |
| 10 | g0pB_taske.txt | orig_taske.txt | 0.34 | 0.11 | 0.15 | 0.49 |
| 11 | g0pC_taska.txt | orig_taska.txt | 0.34 | 0.05 | 0.00 | 0.00 |
| 12 | g0pC_taskb.txt | orig_taskb.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 13 | g0pC_taskc.txt | orig_taskc.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 14 | g0pC_taskd.txt | orig_taskd.txt | 0.85 | 0.42 | 0.34 | 0.97 |
| 15 | g0pC_taske.txt | orig_taske.txt | 0.56 | 0.18 | 0.15 | 0.89 |
| 16 | g0pD_taska.txt | orig_taska.txt | 0.85 | 0.39 | 0.19 | 1.00 |
| 17 | g0pD_taskb.txt | orig_taskb.txt | 0.56 | 0.08 | 0.02 | 0.76 |
| 18 | g0pD_taskc.txt | orig_taskc.txt | 0.34 | 0.22 | 0.20 | 0.58 |
| 19 | g0pD_taskd.txt | orig_taskd.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 20 | g0pD_taske.txt | orig_taske.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 21 | g0pE_taska.txt | orig_taska.txt | 0.56 | 0.90 | 0.81 | 0.99 |
| 22 | g0pE_taskb.txt | orig_taskb.txt | 0.34 | 0.10 | 0.05 | 0.66 |
| 23 | g0pE_taskc.txt | orig_taskc.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 24 | g0pE_taskd.txt | orig_taskd.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 25 | g0pE_taske.txt | orig_taske.txt | 0.85 | 0.18 | 0.13 | 1.00 |
| 26 | g1pA_taska.txt | orig_taska.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 27 | g1pA_taskb.txt | orig_taskb.txt | 0.34 | 0.02 | 0.00 | 0.00 |
| 28 | g1pA_taskc.txt | orig_taskc.txt | 0.56 | 0.10 | 0.00 | 0.26 |
| 29 | g1pA_taskd.txt | orig_taskd.txt | 0.85 | 0.18 | 0.12 | 0.34 |
| 30 | g1pA_taske.txt | orig_taske.txt | 0.05 | 0.01 | 0.00 | 0.00 |
| 31 | g1pB_taska.txt | orig_taska.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 32 | g1pB_taskb.txt | orig_taskb.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 33 | g1pB_taskc.txt | orig_taskc.txt | 0.34 | 0.14 | 0.03 | 0.32 |
| 34 | g1pB_taskd.txt | orig_taskd.txt | 0.56 | 0.09 | 0.03 | 0.35 |
| 35 | g1pB_taske.txt | orig_taske.txt | 0.85 | 0.22 | 0.16 | 0.50 |
| 36 | g1pD_taska.txt | orig_taska.txt | 0.56 | 0.09 | 0.05 | 0.34 |
| 37 | g1pD_taskb.txt | orig_taskb.txt | 0.85 | 0.11 | 0.10 | 0.88 |
| 38 | g1pD_taskc.txt | orig_taskc.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 39 | g1pD_taskd.txt | orig_taskd.txt | 0.05 | 0.02 | 0.06 | 0.00 |
| 40 | g1pD_taske.txt | orig_taske.txt | 0.34 | 0.02 | 0.02 | 0.00 |
| 41 | g2pA_taska.txt | orig_taska.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 42 | g2pA_taskb.txt | orig_taskb.txt | 0.34 | 0.07 | 0.03 | 0.00 |
| 43 | g2pA_taskc.txt | orig_taskc.txt | 0.56 | 0.41 | 0.47 | 0.78 |
| 44 | g2pA_taskd.txt | orig_taskd.txt | 0.85 | 0.22 | 0.25 | 0.31 |
| 45 | g2pA_taske.txt | orig_taske.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 46 | g2pB_taska.txt | orig_taska.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 47 | g2pB_taskb.txt | orig_taskb.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 48 | g2pB_taskc.txt | orig_taskc.txt | 0.34 | 0.00 | 0.07 | 0.00 |
| 49 | g2pB_taskd.txt | orig_taskd.txt | 0.56 | 0.57 | 0.58 | 0.93 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 50 | g2pB_taske.txt | orig_taske.txt | 0.85 | 0.50 | 0.38 | 1.00 |
| 51 | g2pC_taska.txt | orig_taska.txt | 0.85 | 0.34 | 0.45 | 0.66 |
| 52 | g2pC_taskb.txt | orig_taskb.txt | 0.05 | 0.01 | 0.00 | 0.00 |
| 53 | g2pC_taskc.txt | orig_taskc.txt | 0.05 | 0.02 | 0.00 | 0.00 |
| 54 | g2pC_taskd.txt | orig_taskd.txt | 0.34 | 0.15 | 0.16 | 0.56 |
| 55 | g2pC_taske.txt | orig_taske.txt | 0.56 | 0.04 | 0.05 | 0.00 |
| 56 | g2pE_taska.txt | orig_taska.txt | 0.34 | 0.31 | 0.26 | 0.30 |
| 57 | g2pE_taskb.txt | orig_taskb.txt | 0.56 | 0.13 | 0.02 | 0.62 |
| 58 | g2pE_taskc.txt | orig_taskc.txt | 0.85 | 0.00 | 0.00 | 0.78 |
| 59 | g2pE_taskd.txt | orig_taskd.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 60 | g2pE_taske.txt | orig_taske.txt | 0.05 | 0.01 | 0.00 | 0.00 |
| 61 | g3pA_taska.txt | orig_taska.txt | 0.05 | 0.01 | 0.03 | 0.00 |
| 62 | g3pA_taskb.txt | orig_taskb.txt | 0.34 | 0.06 | 0.02 | 0.00 |
| 63 | g3pA_taskc.txt | orig_taskc.txt | 0.56 | 0.27 | 0.22 | 0.55 |
| 64 | g3pA_taskd.txt | orig_taskd.txt | 0.85 | 0.94 | 0.62 | 1.00 |
| 65 | g3pA_taske.txt | orig_taske.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 66 | g3pB_taska.txt | orig_taska.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 67 | g3pB_taskb.txt | orig_taskb.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 68 | g3pB_taskc.txt | orig_taskc.txt | 0.34 | 0.07 | 0.03 | 0.00 |
| 69 | g3pB_taskd.txt | orig_taskd.txt | 0.56 | 0.08 | 0.00 | 0.30 |
| 70 | g3pB_taske.txt | orig_taske.txt | 0.85 | 0.24 | 0.19 | 1.00 |
| 71 | g3pC_taska.txt | orig_taska.txt | 0.85 | 0.38 | 0.14 | 0.99 |
| 72 | g3pC_taskb.txt | orig_taskb.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 73 | g3pC_taskc.txt | orig_taskc.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 74 | g3pC_taskd.txt | orig_taskd.txt | 0.34 | 0.11 | 0.00 | 0.52 |
| 75 | g3pC_taske.txt | orig_taske.txt | 0.56 | 0.09 | 0.00 | 0.78 |
| 76 | g4pB_taska.txt | orig_taska.txt | 0.05 | 0.01 | 0.00 | 0.00 |
| 77 | g4pB_taskb.txt | orig_taskb.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 78 | g4pB_taskc.txt | orig_taskc.txt | 0.34 | 0.27 | 0.21 | 0.61 |
| 79 | g4pB_taskd.txt | orig_taskd.txt | 0.56 | 0.28 | 0.17 | 0.82 |
| 80 | g4pB_taske.txt | orig_taske.txt | 0.85 | 0.55 | 0.41 | 0.93 |
| 81 | g4pC_taska.txt | orig_taska.txt | 0.85 | 0.90 | 0.77 | 0.89 |
| 82 | g4pC_taskb.txt | orig_taskb.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 83 | g4pC_taskc.txt | orig_taskc.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 84 | g4pC_taskd.txt | orig_taskd.txt | 0.34 | 0.80 | 0.85 | 0.97 |
| 85 | g4pC_taske.txt | orig_taske.txt | 0.56 | 0.36 | 0.40 | 0.91 |
| 86 | g4pD_taska.txt | orig_taska.txt | 0.56 | 0.09 | 0.10 | 0.28 |
| 87 | g4pD_taskb.txt | orig_taskb.txt | 0.85 | 0.00 | 0.00 | 0.93 |
| 88 | g4pD_taskc.txt | orig_taskc.txt | 0.05 | 0.01 | 0.00 | 0.00 |
| 89 | g4pD_taskd.txt | orig_taskd.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 90 | g4pD_taske.txt | orig_taske.txt | 0.34 | 0.15 | 0.08 | 0.51 |
| 91 | g4pE_taska.txt | orig_taska.txt | 0.34 | 0.01 | 0.00 | 0.00 |
| 92 | g4pE_taskb.txt | orig_taskb.txt | 0.56 | 0.35 | 0.35 | 0.93 |
| 93 | g4pE_taskc.txt | orig_taskc.txt | 0.85 | 0.16 | 0.26 | 0.32 |
| 94 | g4pE_taskd.txt | orig_taskd.txt | 0.05 | 0.00 | 0.00 | 0.00 |
| 95 | g4pE_taske.txt | orig_taske.txt | 0.05 | 0.02 | 0.04 | 0.00 |

**APPENDIX B:** The Clough-Stevenson's classification of the level of plagiarism (Plg.) in documents

| Documents | Plg. | Documents | Plg. | Documents | Plg. | Documents | Plg. |
|---|---|---|---|---|---|---|---|
| g0pA_taska.txt | non | g0pE_taske.txt | cut | g2pB_taskd.txt | light | g3pC_taskc.txt | non |
| g0pA_taskb.txt | cut | g1pA_taska.txt | non | g2pB_taske.txt | cut | g3pC_taskd.txt | heavy |
| g0pA_taskc.txt | light | g1pA_taskb.txt | heavy | g2pC_taska.txt | cut | g3pC_taske.txt | light |
| g0pA_taskd.txt | heavy | g1pA_taskc.txt | light | g2pC_taskb.txt | non | g4pB_taska.txt | non |
| g0pA_taske.txt | non | g1pA_taskd.txt | cut | g2pC_taskc.txt | non | g4pB_taskb.txt | non |
| g0pB_taska.txt | non | g1pA_taske.txt | non | g2pC_taskd.txt | heavy | g4pB_taskc.txt | heavy |
| g0pB_taskb.txt | non | g1pB_taska.txt | non | g2pC_taske.txt | light | g4pB_taskd.txt | light |
| g0pB_taskc.txt | cut | g1pB_taskb.txt | non | g2pE_taska.txt | heavy | g4pB_taske.txt | cut |
| g0pB_taskd.txt | light | g1pB_taskc.txt | heavy | g2pE_taskb.txt | light | g4pC_taska.txt | cut |
| g0pB_taske.txt | heavy | g1pB_taskd.txt | light | g2pE_taskc.txt | cut | g4pC_taskb.txt | non |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| g0pC_taska.txt | heavy | g1pB_taske.txt | cut | g2pE_taskd.txt | non | g4pC_taskc.txt | non |
| g0pC_taskb.txt | non | g1pD_taska.txt | light | g2pE_taske.txt | non | g4pC_taskd.txt | heavy |
| g0pC_taskc.txt | non | g1pD_taskb.txt | cut | g3pA_taska.txt | non | g4pC_taske.txt | light |
| g0pC_taskd.txt | cut | g1pD_taskc.txt | non | g3pA_taskb.txt | heavy | g4pD_taska.txt | light |
| g0pC_taske.txt | light | g1pD_taskd.txt | non | g3pA_taskc.txt | light | g4pD_taskb.txt | cut |
| g0pD_taska.txt | cut | g1pD_taske.txt | heavy | g3pA_taskd.txt | cut | g4pD_taskc.txt | non |
| g0pD_taskb.txt | light | g2pA_taska.txt | non | g3pA_taske.txt | non | g4pD_taskd.txt | non |
| g0pD_taskc.txt | heavy | g2pA_taskb.txt | heavy | g3pB_taska.txt | non | g4pD_taske.txt | heavy |
| g0pD_taskd.txt | non | g2pA_taskc.txt | light | g3pB_taskb.txt | non | g4pE_taska.txt | heavy |
| g0pD_taske.txt | non | g2pA_taskd.txt | cut | g3pB_taskc.txt | heavy | g4pE_taskb.txt | light |
| g0pE_taska.txt | light | g2pA_taske.txt | non | g3pB_taskd.txt | light | g4pE_taskc.txt | cut |
| g0pE_taskb.txt | heavy | g2pB_taska.txt | non | g3pB_taske.txt | cut | g4pE_taskd.txt | non |
| g0pE_taskc.txt | non | g2pB_taskb.txt | non | g3pC_taska.txt | cut | g4pE_taske.txt | non |
| g0pE_taskd.txt | non | g2pB_taskc.txt | heavy | g3pC_taskb.txt | non | | |

# Domain Specific Named Entity Recognition Using Supervised Approach

**Ashwini A. Shende** *zashwini@rediffmail.com*
*Department of Computer Science & Engineering, RCOEM,*
*Rashtrasant Tukdoji Maharaj, Nagpur University*
*Nagpur, 440013, India*

**Avinash J. Agrawal** *avinashjagrawal@gmail.com*
*Department of Computer Science & Engineering, RCOEM,*
*Rashtrasant Tukdoji Maharaj, Nagpur University*
*Nagpur, 440013, India*

**Dr. O. G. Kakde** *ogkakde@vnit.ac.in*
*Visvesvaraya National Institute of Technology*
*Nagpur, 440010, India*

**Abstract**

This paper introduces Named Entity Recognition approach for text corpus. Supervised Statistical methods are used to develop our system. Our system can be used to categorize NEs belonging to a particular domain for which it is being trained. As Named Entities appears in text surrounded by contexts (words that are left or right of the NE*)*, we will be focusing on extracting NE contexts from text and then performing statistical computing on them. We are using n-gram model for extracting contexts from text. Our methodology first extracts left and right tri-grams surrounding NE instances in the training corpus and calculate their probabilities. Then all the extracted tri-grams along with their calculated probabilities are stored in a file. During testing, system detects unrecognized NEs from the testing corpus and categorizes them using the tri-gram probabilities calculated during training time. The proposed system is made up of two modules i.e. Knowledge acquisition and NE Recognition. Knowledge acquisition module extracts tri-grams surrounding NEs in the training corpus and NE Recognition module performs the categorization of unrecognized NEs in the testing corpus.

**Keywords:** Named Entity, Supervised Machine learning, N-gram, Context Extraction, NE Recognition

## 1. INTRODUCTION

The term "Named Entity" (NE) is frequently used in Information Extraction (IE) applications. It was coined at the sixth Message Understanding Conference (MUC-6) which influenced IE research in the 1990s. In defining IE tasks, people noticed that it is essential to recognize information units such as names including person, organization, and location names, and numeric expressions including time, date, money, and percentages. Identifying references to these entities in text was acknowledged as one of IE's important sub-tasks and was called "Named Entity Recognition (NER)." Named Entity Recognition is complex in various areas of automatic Natural Language Processing of (NLP), document indexing, document annotation, translation, etc. It is a fundamental step in various Information Extraction (IE) tasks.

### 1.1 Named Entity Recognition

The NER task consists of identifying the occurrences of some predefined phrase types in a text. In the expression "Named Entity," the word "Named" aims to restrict the task to only those entities for which one or many rigid designators, stands for the referent. Some tasks related to NER (David Nadeau et.al. [1]) can be listed as follows.

- **Personal Name Disambiguation :**

It is the task of identifying the correct referent of a given designator. In a given context, it may consist of identifying whether *Jim Clark* is the race driver, the film editor, or the Netscape founder. Corpus-wide disambiguation of personal names has applications in document clustering for information retrieval.

- **NE Descriptions Identification :**

It is the identification of textual passages that describe a given NE. For instance, Bill Clinton is described as "the President of the U.S.," "the democratic presidential candidate" or "an Arkansas native," depending on the document. Description identification can be use as a clue in personal name disambiguation.

- **Named Entity Translation :**

It is the task of translating NEs from one language to another. For instance, the French translation of "National Research Council Canada" is "Camseil national de recherché Canada." NE translation is acknowledged as a major issue in machine translation.

- **Analysis of Name Structu**

It is the identification of the parts in a person name. For example, the name "Doctor Paul R. Smith" is composed of a person title, a first name, a middle name, and a surname. It is presented as a preprocessing step for NER and for the resolution of co-references to help. Determine, for instance, that "John F. Kennedy" and "President Kennedy" is the same person, while "John F. Kennedy" and "Caroline Kennedy" are two distinct persons.

- **Entity Anaphora Resolution :**

It mainly consists of resolving pronominal co-reference when the antecedent is an NE. For example, in the sentence "Rabi finished reading the book and he replaced it in the library," the pronoun "he" refers to "Rabi." Anaphora resolution can be useful in solving the NER problem itself by enabling the use of extended co-reference networks. Meanwhile it has many applications of its own, such as in "question answering" (e.g., answering "Who put the book in the library?").

- **Acronym Identification :**

It is described as the identification of an acronym's definition (e.g., "IBM" stands for "International Business Machines") in a given document. The problem is related to NER because many organization names are acronyms (GE, NRC, etc.). Resolving acronyms is useful, again, to build co-reference networks aimed at solving NER On its own; it can improve the recall of information retrieval by expanding queries containing an acronym with the corresponding definition.

- **Record linkage :**

It is the task of matching named entities across databases. It involves the use of clustering and string matching techniques in order to map database entries having slight variations. It is used in database cleaning and in data mining on multiple databases.

- **Case Restoration :**

It consists of restoring expected word casing in a sentence. Given a lower case sentence, the goal is to restore the capital letters usually appearing on the first word of the sentence and on NEs. This task is useful in machine translation, where a sentence is usually translated without capitalization information.

Computational research aiming at automatically identifying NEs in texts forms a vast and heterogeneous pool of strategies, methods, and representations. In its canonical form, the input of an NER system is a text and the output is information on boundaries and types of NEs found in the text. The majority of NER systems fall in two categories: the Rule**-based** systems; and the **Statistical** systems. While early studies were mostly based on handcrafted rules, most of the recent systems preferred statistical methods. In both approaches, large collections of documents are analyzed by hand to obtain sufficient knowledge for designing rules or for feeding machine learning algorithms. Expert linguists must execute this important amount of work, which in turn limits the building and maintenance of large-scale NER systems.

The ability to recognize previously unknown entities is an essential part of NER systems. Such ability hinges upon recognition and classification rules triggered by distinctive modeling features

associated with positive and negative examples. When training examples are not available, handcrafted rules systems remain the preferred technique. The statistical methods collect statistical knowledge from corpus and determine NE categories based on the statistical knowledge. The statistical methods use supervised machine learning algorithms. The idea of supervised learning is to study the features of positive and negative examples of NE over a large collection of annotated documents and design rules that capture instances of a given type. The main shortcoming of Supervised Learning is the requirement of a large annotated corpus. The unavailability of such resources and the prohibitive cost of creating them lead to two alternative learning methods: semi-supervised learning **(SSL**); and unsupervised learning (**UL**).

The term "**semi-supervised**" or "**weakly supervised**" is relatively recent. The main technique for SSL is called "bootstrapping" and involves a small degree of supervision, such as a set of seeds, for starting the learning process. For example, a system aimed at "disease names" might ask the user to provide a small number of example names. Then, the system searches for sentences that contain these names and tries to identify some contextual clues common to the five examples. Then, the system tries to find other instances of disease names appearing in similar contexts. The learning process is then reapplied to the newly found examples, so as to discover new relevant contexts. By repeating this process, a large number of disease names and a large number of contexts will eventually be gathered.

The typical approach in **unsupervised** learning is clustering. For example, one can try to gather NEs from clustered groups based on context similarity. There are other unsupervised methods also. Basically, the techniques rely on lexical resources (e.g., WordNet), on lexical patterns, and on statistics computed on a large unannotated corpus.

This paper discusses the use of supervised machine learning approach for the problem of NE recognition. The aim of our study is to reveal contextual NE in a document corpus using n-gram modeling. A context considers words surrounding the NE in the sentence in which it appears, it is a sequence of words, that are left or right of the NE. In this work, we use supervised learning technologies, combined with statistical models to extract contexts from text document corpus, to identify the most pertinent contexts for the recognition of a NE.

### 1.2 n-gram Modeling

A useful part of the knowledge needed for Word Prediction can be captured using simple statistical techniques like the notion of the probability of a sequence (a phrase, a sentence). An ***n*-gram model** is a type of probabilistic model for predicting the next item in a sequence. n-gram probabilities can be used to estimate the likelihood

- Of a word occurring in a context (n-1)
- Of a sentence occurring at all

*n*-gram models are used in various areas of statistical natural language processing and genetic sequence analysis. n-gram language model uses the previous n-1 words in a sequence to predict the next word. These models are trained using very large corpora. n-gram probabilities come from a training corpus

- overly narrow corpus: -probabilities don't generalize
- overly general corpus:- probabilities don't reflect task or domain

A separate test corpus is used to evaluate the model, typically using standard metrics

- held out test set; development test set
- cross validation
- results tested for statistical significance

An ***n*-gram** is a subsequence of *n* items from a given sequence. The items can be phonemes, syllables, letters, words or base pairs according to the application. An *n*-gram of size 1 is referred to as a "**unigram**"; size 2 is a "**bigram**" (or, less commonly, a "digram"); size 3 is a "**trigram**"; size 4 is a "**four-gram**" and size 5 or more is simply called an "***n*-gram**"...

E.g. for the sequence "the big red ball"

unigram P (**ball)**
bigram P **(ball** / **red)**
trigram P **(ball** / **big red)**
four-gram P **(ball** / **the big red)**

In general
**P (Word| Some fixed prefix)**

As we increase the value of n, the accuracy of n-gram model increases, since choice of next word becomes increasingly constrained.

n-gram is a sequence of *n* words in a text document and one can get a set of n-grams by moving a floating window from the beginning to the end of the document. During the n-gram extraction from text document, duplicate n-grams must be removed and the frequency of the n-gram types should be calculated. Additionally, other values can be stored with n-gram type and frequency, e.g. n-gram unique number, but it is document and query model dependent.

FIGURE 1, shows a common architecture of an n-gram extraction framework. This framework usually includes:

1. **Document parsing** – it parses terms from input documents.
2. **Term pre-processing** – in this phase, various techniques like stemming and stop-list are applied for the reduction of terms.
3. **n-gram building and pre-processing** – it creates an n-gram as a sequence of *n* terms. Sometimes, n-grams are not shared by text units (sentences or paragraphs). It means, the last term of a sentence is the last term of an n-gram and the next n-gram begins by the first term of the next sentence.
4. **n-gram extraction** – the main goal of this phase is to remove duplicate n-grams. The result of this phase is a collection of n-gram types with the frequency enclosed to each type. This collection can be cleaned after this phase; for example, n-gram types with a low frequency are removed. However, it is not appropriate to apply this post-processing in any application. It can be used only when we do not need low frequency n-gram types. A common part of such a framework is n-gram indexing. A data structure is applied to speed up access to the tuple _ngram, id, frequency_, where ngram is a key; it means the ngram is an input of the query and id and frequency form the output. Although, it is necessary to create other data structures, for specific document and query models, one must always consider this global storage of the tuples.

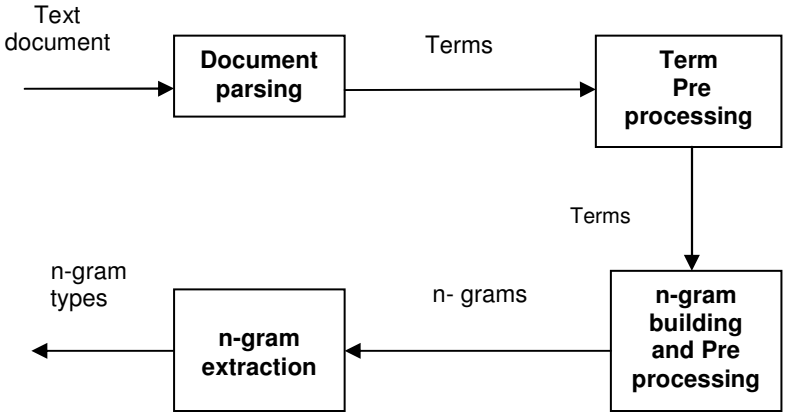



**FIGURE 1:** n-gram Extraction Framework

The remaining of the paper is organized as follows: Section 2 presents the review of the various methods used for Named Entity Recognition. Section 3 describes the Methodology and section 4 gives test results of our approach. Section 5 gives the Work's conclusion and Section 6 explains the future work recommended.

## 2. RELATED WORK

Named entity recognition can be used to perform numerous processing tasks in various areas: of Information Extraction systems, Text mining, Automatic Speech Recognition (ASR) etc. Several works are particularly interested in the recognition of named entities.

Mikheev et al. [2] have built a system for recognizing named entities, which combines a model based on grammar rules, and statistical models, without resorting to named entity lists.

Collins et al. [3] suggests an algorithm for named entity classification, based on the meaning word disambiguation, and exploits the redundancy in the contextual characteristics. This system operates a large corpus to produce a generic list of proper nouns. The names are collected by searching for a syntax diagram with specific properties. For example, a proper name is a sequence of consecutive words in a nominal phrase, etc.

Petasis et al. [4] presented a method that helps to build a rules-based system for recognition and classification of named entities. They have used machine learning, to monitor system performance and avoid manual marking.

In his paper, Mann et al. [5] explores the idea of fine-grained proper noun ontology and its use in question answering. The ontology is built from unrestricted text using simple textual co-occurrence patterns. This ontology is therefore used on a question answering task to provide primary results on the utility of this information. However, this method has a low coverage.

The Nemesis system presented by Fourour et al.[ 6] is founded on some heuristics, allowing the identification of named entities, and their classification by detecting the boundaries of the entity called "context" to the left or right, and by studying syntactic, or morphological nature of these entities. (n-gram modeling) For example, acronyms are named entities consisting of a single lexical unit comprising several capital letters, etc.

Krstev et al. [7] suggested a basic structure of a relational model of a multilingual dictionary of proper names based on four-level ontology.

Etzioni et al. [8] planned the KNOWITALL system which aims at automating the process of extracting named entities from the Web in an unsupervised and scalable manner. This system is not intended for recognizing a named entity, but used to create long lists of named entities. However, it is not designed to resolve the ambiguity in some documents.

Friburger et al. [9] recommends a method based on rules for finding a large proportion of person names. However, this method has some limitations as errors, and missing responses.

Nadeau et al. [10] have suggested a system for recognizing named entities. Their work is based on those of Collins, and Etzioni. The system exploits human-generated HTML markup in Web pages to generate gazetteers, then it uses simple heuristics for the entity disambiguation in the context of a given document.

Kono Kim et. al. [11] proposed a NE (Named Entity) recognition system using a semi supervised statistical method. In training time, the NE recognition system builds error-prone training data only using a conventional POS (Part-Of-Speech) tagger and a NE dictionary that semi-automatically is constructed. Then, the NE recognition system generates a co-occurrence similarity matrix from

the error prone training corpus. In running time, the NE recognition system detects NE candidates and assigns categories to the NE candidates using Viterbi searching on the AWDs.

In view of works touching the recognition of named entities, we perceive that most of them are based on a set of rules in relation to predefined categories: morphological, grammatical, etc. or on predefined lists or dictionaries. The n-gram modeling domain is still in exploration. We adopted the idea of Nemesis based on the left and the right context of the named entity. However, our approach does not mark the context derived from syntactic or morphological rules, but identifies the context founded on learning phase. The objective is thus to carry out a system, able to induce the nature of a named entity, without requiring dictionaries or lists of named entities.

## 3. METHADOLOGY

This paper discusses the use of supervised machine learning approach for the problem of NE recognition. The aim of our study is to reveal contextual NE in a document corpus using n-gram modeling. A context consists of words surrounding the NE in the sentence in which it appears. It is a sequence of words, that are left or right of the NE. In this work, we use supervised learning technologies, combined with statistical methods to extract contexts from text document and to identify the most pertinent contexts for the recognition of a NE.

Our work mainly focuses on Context extraction i.e. extracting the left and the right context of the Named entity... Two or more words that tend to occur in similar linguistic context (i.e. to have similar Co-occurrence pattern), tend to be positioned closer together in semantic space and tend to resemble each other in meaning. Our objective is to carry out a system, able to induce the nature of a named entity, following the meeting of certain indicators.

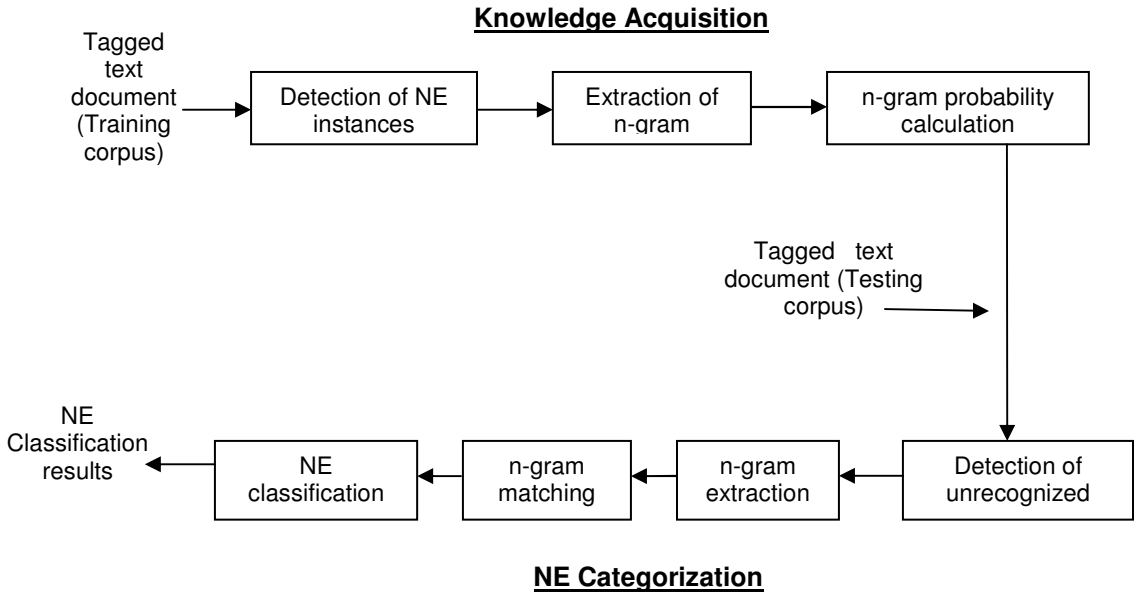



**FIGURE 2:** Block diagram of Proposed System

FIGURE. 2 shows block diagram of the proposed system. The proposed system consists of two modules. First module is a Knowledge acquisition module which detects NE instances from the training corpus. Then, it extracts left and right tri-grams surrounding those NE instances and calculates its probability occurrence in the training corpus. After calculating all probabilities, extracted tri-grams along with their probabilities are stored in a text file for reference. When testing corpus is given for testing, NE recognition module finds all unrecognized NE instances from it by using the same method used in knowledge acquisition module. Then, it classifies each

unrecognized NE instance in the testing corpus into one of the domain specific categories using the tri-gram probabilities already stored in a file.

### 3.1. Knowledge Acquisition

The main functioning of this module is to extract the tri-grams surrounding NEs from given domain specific text document. The document acts as the training corpus for learning. Our system input is a tagged text document. For our corpus, all NEs should have numerical tagging. Some of the sentences from our corpus are given below.

- when [q] vidarbha [1] express [n] reaches [v] wardha [2]
- what [q] is [x] the [d] status [n] of [p] mumbai [1] mail [n]
- what [q] is [x] departure [n] time [n] of [p] vidarbha [1] express [n]
- when [q] mumbai [1] mail [n] reaches [v] mumbai [2]
- what [q] is [x] the [d] position [n] of [p] the [d] gitanjali [1] express [n]

**ALGORITHM:-**

- Locate all NEs from training corpus.
- Extract left & right trigrams surrounding NEs.
    - If trigram does not exist then extract bigrams.
        - If bigram does not exist then extract unigrams.
- Remove duplicate trigrams / bigrams / unigrams and calculate the probability of each in the corpus.
- Store the unique trigrams / bigrams / unigrams along with probability in a file.

The first step of our algorithm is to locate Named Entities in each sentence by reading the text corpus. NE's are the words which are followed by numerical tagging. E.g. "*vidarbha* ", "*Mumbai*","*wardha*" etc are NE instances in the above examples. After locating the NEs, surrounding trigrams are extracted from the text corpus. Trigrams are the 3 consecutive words to the left or right of NE. For efficiency purpose we will extract both left and right trigrams for each NE. following structure is used to store the trigram.

```
public class TriGramElement
{

    public String[] LeftElements = new String[3];
    public String[] RightElements = new String[3];
    public String CentreElement;
    public String[] LeftValue = new String[3];
    public String[] RightValue = new String[3];
    public String CentreValue;

};
```

Every NE occurrence cannot guarantee presence of trigram surrounding it, especially if NE occurs as the first or last word of the sentence in a corpus. In such cases our system is flexible to consider either Bigram or unigram. E.g. for NE "*vidarbha*" left context is a unigram and right context consists of a trigram. For "*Mumbai*", left context is trigram and right content is unigram and for "*hawrah*" both left and right context consists of trigrams. Some sample extracted trigrams from the corpus is mentioned below.

when **vidarbha** express reaches wardha

the status of **mumbai** mail

by what time **hawrah** mail will come

The next step of the algorithm is to remove duplicate n-grams. Removal of duplicate trigrams is necessary to apply statistical methods on it. For probability calculation we need to get the occurrence count of each trigram. Our system generates a list of unique trigrams and stores them in a text file along with their probabilities. The sample trigrams stored in a text file is shown below.

```
1    ,,,when          :   wardha,reaches,express    -->  0.02

1    ,the,status,of   :   is,what,mail              ->  0.02

1  ,position,of,the   :   is,what,express           -->  0.02

3  ,the,fare,from     :   what,gondia,to            -->  0.02
```

**FIGURE 3:** List of sample trigrams stored in a file

### 3.2 NE Categorization

After detecting unrecognized NEs, the NE recognition module assigns categories to them using the trigram probabilities calculated by Knowledge acquisition module.

**ALGORITHM:**

- Detect unrecognized NE instance from testing corpus.
- Extract left and right trigrams for it.
    - If trigram does not exist then extract bigram.
        - If bigram does not exist then extract unigram.
- For every unrecognized NE instance in testing corpus, search for left trigram / bigram / unigram in the list stored in a file. (Generated from training corpus) using linear search.
- If match not found search for right trigram / bigram / unigram in the list.
- If match not found for left as well as right trigram / bigram / unigram then marked the corresponding NE as unrecognized.
- Find out the category of maximum probability trigram / bigram / unigram match.
- Assign maximum probability category to the unrecognized NE.
- Repeat above steps for all unrecognized NE instances in the testing corpus.
- Store NE categorization results in a file.

NE categorization module will first extract all NE instances, from the testing corpus by applying the same method used in knowledge acquisition module. We are assuming that testing corpus is a tagged corpus in which all unrecognized NE's are marked with tag [0]. After detecting NE's, NE categorization module will create a list of unrecognized NE instances. For each NE stored in list, left and right content words are extracted from the testing corpus in the form of trigrams.

To categorize NE, our system will compare its left context words with the tri-gram entries (generated from training corpus) stored in a file using linear search algorithm. Our system prefers left context words over right context words as left context is more relevant in comparison to right context for recognition. If the match is found then its probability count value will be extracted. After checking all the entries, NE categorization module will compare probability count of all matched entries and will find the maximum probability count out of it. Then unrecognized NE will be classified to the matched category for which probability count is maximum. In absence of left trigrams, right trigrams will be considered for matching. Our system is flexible to use bigrams as well as unigrams in absence of trigrams for categorizing NEs. After categorizing all NE's categorization results will be stored in a text file.

Consider the following sentence from the testing corpus

how [q] many [u] trains [n] are [x] of [p] type [n] **doronto** [0]

In the above sentence word with tag [0] is detected as unrecognized NE. i.e. "*doronto*". Next step is to find out the context words to the left and right of NE. The extracted tri-gram for the "*doronta*" is

are of type **doronto** --- ---- -----

In above case left trigram consists of 3 words whereas right tri-gram is null as "*doronto*" is the last word of the sentence. Our algorithm gives precedence to left context words .So it will search the Tri-gram entries stored in a file to get a match for tri-gram "*are of type* ". The match is found with probability count 0.02 and the category type is train name. So "*doronto*" will be categorized to Train name and result will be stored in a text file.

## 4. EXPERIMENTAL RESULTS

### 4.1 Test Collections

To evaluate the performance of the proposed system, we used a test collection of a Railway Reservation domain. The testing corpus is a collection of routine railway enquiries consisting of domain specific NE categories like train names, source and destination train names, reservation classes etc. Categories are labeled with numerical tagging in the testing corpus. We think that the preliminary experiments have some meaning as our goal is to recognize NE categories with supervised statistical methods.

### 4.2 Performance Evaluation

Since any NER system or method must produce a single, unambiguous output for any Named Entity in the text, the evaluation is not based on a system architecture in which Named Entity Recognition would be completely handled as a preprocess to sentence and discourse analysis. The task requires that the system recognize what a NE represents, not just its superficial appearance and the answer may have to be obtained using techniques that draw information from a larger context or from reference lists.

A scoring model developed for the MUC and Named Entity Task evaluations measures, both precision (*P*) and recall (*R*) terms borrowed from the information-retrieval community. These two measures of performance combine to form one measure of performance, the *F*-measure, which is computed by the uniformly weighted harmonic mean of precision and recall.

To evaluate performance of the proposed system, we used the performance measures like precision, recall and the F-score. Precision (p) is the proportion of correct responses out of

returned NE categories, and Recall (r) is the proportion of returned NE categories out of classification targets. Following graph shows the performance measure results of our system.

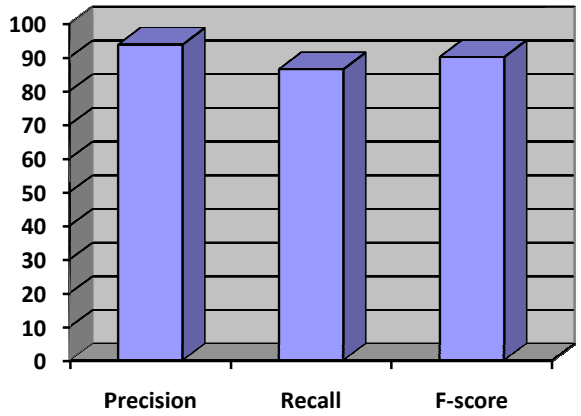



**FIGURE 4:** Performance Measure results

## 5. CONCLUSION

We proposed a NE recognition system using Supervised Statistical methods. Our goal is to uncover Named Entity in a document corpus. NE occurs frequently accompanied by contexts: i.e. sequence of words, that are left or right of the NE. In training time, the proposed system extracts all NE instances from a given domain specific text document. Then, the proposed system generates a list of unique tri-grams surrounding NEs in the training corpus and calculate probability occurrence for each. This information is stored in a file as a reference for testing. During testing, this information is referred to identify most pertinent contexts for the categorization of unrecognized NEs from the testing corpus. This enables to derive a model for NE recognition. In the preliminary experiments on Railway Reservation domain the proposed system showed 90.04% average F-score measure**.**

Recall and precision are usually admitted parameters for measuring system performance in the NER field.

$$\textbf{Precision} = \frac{\textbf{( No. Of correct responses )}}{\textbf{( No. of responses )}}$$

$$\textbf{Recall} = \frac{\textbf{(No. Of correct responses )}}{\textbf{(No. correct in key)}}$$

$$\textbf{F- measure} = \frac{\textbf{Precision} \times \textbf{Recall}}{\textbf{½ (Precision + Recall )}}$$

For NER task it is observed that though Hand-made rule based approach can get high rate results in specific domain but it has problem with broad and new domain. As Hand-made rule based method are dependent to domain, Machine learning–based methods is the best independent solution for NER. Machine Learning methods can get good result in precision and recall with high portability and it can be best independent and portable solution for text mining and specially NER. But high performance of this kind of methods depends on the data training value. This type of approach can get high precision in recognition when amount of data training is huge and the result is strictly reduce when data training value is few or malfunction of algorithm. The Hybrid methods gave good results but portability of this type of approach is reduced when they improve precision in recognition by using huge value of fixed rules. Though traditionally Rule based systems were more popular now a days machine learning approach is preferred for developing NER systems. TABLE 1 shows the comparison of the results obtained from proposed systems with the existing systems.

| | System | Precision | Recall | F -Score |
|---|---|---|---|---|
| **1** | **Proposed System** | **93.68** | **86.40** | **90.04** |
| 2 | NYU System (Rule based) | 90 | 86 | 88.19 |
| 3 | IsoQuest,Inc (Rule based) | 93 | 90 | 91.60 |
| 4 | MENE (Machine learning based) | 96 | 89 | 92.20 |
| 5 | Association Rule Mining (Machine learning based) | 83.43 | 66.34 | 70.16 |
| 6 | IdentiFinder (Machine learning based) | 92 | 89 | 90.44 |
| 7 | LTG (hybrid) | 95 | 92 | 93.39 |
| 8 | NYU Hybrid (hybrid) | 93 | 85 | 88.80 |

**TABLE 1:** Comparison of proposed system results with the existing systems

All the proposed methods and models developed for NER task have tried to improve precision in recognition module and portability in recognition domain as one of the major problem and difficulty in NER systems is to change and switch over to a new domain called portability. Most distinguishing feature of the proposed system is that it is easily portable to the new domain as it is based on supervised machine learning approach. Proposed system is using n-gram model for extracting NE context which is also contributing to the portability of the proposed system across multiple domains. It is not required to maintain large gazetteer lists as NE recognition for our system is context based. Context is extracted from the corpus itself (training as well as testing) and not dependent on gazetteer lists. Based on the experimental results it can be said that the proposed system is a good solution to address NER problem as it is capable of recognizing NEs from the given domain corpora dynamically without maintaining huge large NE dictionaries or gazetteer lists.

## 6. FUTURE WORK

Though primarily we have applied the proposed approach to address NER problem, it is not restricted to that problem only. The proposed approach can be applied to solve many problems in Natural Language Processing domain. It can be used in various research areas like machine translation, Question answering systems etc.

As we have stated that proposed system is portable in nature we need to use our systems across diverse domains and get its performance analysis across those diverse domains.

Our future work recommendations are as follows.

- To test the system on different domain corpora,
- To discern and to measure similarity between contexts. We can use this measurement to cluster similar contexts.
- Though we have primarily applied our approach to NER problem, we can also attempt some additional concepts

## 7. REFERENCES

[1] David Nadeau "Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision "

[2] Mikheev, M. Moens, and C. Grover, "Named Entity Recognition without Gazetteers", *in Proceedings of Conference of European, Chapter of the Association for Computational Linguistics, EACL '99,* pp. 1-8, University of Bergen, Bergen, Norway June 1999.

[3] M. Collins and Y. Singer, "Unsupervised models for named entity classification", *in Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999, pp. 189–196*

[4] G Petasis, F Vichot, F Wolinski, G Paliouras, V. Karkaletsis, and C. D. Spyropoulos, "Using machine learning to maintain rule-based named-entity recognition and classification", *in Proceedings of the 39th Annual Meeting on Association for Computational Linguistic*s, pp. 426 – 43, Toulouse, France, 2001

[5] G.S. Mann, "Fine-grained proper noun anthologies for question answering", *International Conference on Computational Linguistics*, *COLING-02 on SEMANET: building and using semantic networks, 2002, Vol. 11*,

[6] N. Fourour, and E.Morin, "Apport du Web dans la reconnaissance des entités nommées". *Revue québécoise de linguistique, 2003, vol. 32, n° 1, pp. 41-60.*

[7] Krstev, D. Vitas, D. Maurel, M. Tran, "Multilingual ontology of proper name", *in Proceedings of the Language and Technology Conference, pp. 116–119, Poznan, Poland, 2005*

[8] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A.Yates, "Unsupervised named-entity extraction from the web: An experimental study", *Artificial Intelligence*, *2005, vol. 65,pp. 91–134*

[9] N. Friburger, "Linguistique et reconnaissance automatique des noms propres", *Meta : journal des traducteurs,2006, vol. 51, n° 4, pp. 637-650*

[10] David Nadeau, Peter D. Turney and Stan Matwin "Unsupervised Named-Entity Recognition: Generating Gazetteers and Resolving Ambiguity", In Proceedings of the *19th Canadian Conference on Artificial Intelligence*, *2006*

[11] Kono Kim, Yeohoon Yoon , Harksoo Kim, and Jungyun Seo ",Named Entity Recognition Using Acyclic Weighted Digraphs: A Semi-supervised Statistical Method", *PAKDD 2007, LNAI 4426, pp. 571–578, 2007. © Springer-Verlag Berlin Heidelberg 2007*

# Dictionary Entries for Bangla Consonant Ended Roots in Universal Networking Language

**Mohammad Zakir Hossain Sarker** *zakir.publications@gmail.com*
*Department of CSE*
*Jahangirnagar University*
*Savar, Dhaka, Bangladesh*

**Md. Nawab Yousuf Ali** *nawab@ewubd.edu*
*Department of CSE*
*East West University*
*Dhaka, Bangladesh*

**Jugal Krishna Das** *drdas64@yahoo.com*
*Department of CSE*
*Jahangirnagar University*
*Savar, Dhaka, Bangladesh*

**Abstract**

The Universal Networking Language (UNL) deals with the communication across nations of different languages and involves with many different related discipline such as linguistics, epistemology, computer science etc. It helps to overcome the language barrier among people of different nations to solve problems emerging from current globalization trends and geopolitical interdependence. We are working to include Bangla language in the UNL system so that Bangla language can be converted to UNL expressions. As a part of this process currently we are working on Bangla Consonant Ended Verb Roots and trying to develop lexical or dictionary entries for the Consonant Ended Verb Roots. In this paper, we have presented our work by describing Bangla verb, Verb root, Verbal Inflections and then finally showed the dictionary entries for the consonant ended roots.

**Keywords:** Universal Networking Language, Verb Root, Consonant Ended Verb Root, Vowel Ended Verb Root, Verbal Inflections, Dictionary Entry

## 1. INTRODUCTION

Universal Networking Language (UNL) is a declarative formal language specifically designed to represent semantic data extracted from natural language texts. It can be used as a pivot language in inter-lingual machine translation systems or as a knowledge representation language in information retrieval applications. Currently, the UNL includes 16 languages [1], which are the six official languages of the United Nations (Arabic, Chinese, English, French, Russian and Spanish), in addition to the ten other widely spoken languages (German, Hindi, Italian, Indonesian, Japanese, Latvian, Mongol, Portuguese, Swahili and Thai). In the last few years, machine translation techniques have been applied to web environments. The growing amount of available multilingual information on the Internet and the Internet users has led to a justifiable interest on this area. Hundreds of millions of people of almost all levels of education, attitudes and different jobs all over the world use the Internet for different purposes [2], where English is the main language of the Internet. But English is not understandable for most of the people. Interlingua translation programs are needed to develop. The main goal of the UNL system, which allows users to visualize websites in their native languages, is to provide a common representation for accessing Internet of multilingual websites by the majority of the people over the world. For this common representation, lexical (dictionarical) knowledge is a critical issue in

natural language processing systems, where the development of large-scale lexica with specific formats capable of being used by distinguished applications, in particular to multilingual systems, has been given special focus. Our goal is to include Bangla in this system with less effort. To do so we have working on UNL, Bangla Grammar and then the step by step process for including Bangla in the UNL system. As a part of the process, in this paper, we have explained and described Bangla Verb, Verb Root especially Consonant Ended Verb Root, Verbal Inflections, etc. and dictionary entries of Consonant Ended Verb Roots and their Verbal Inflections.
The organization of this paper is as follow: In Section 2 we describe the Research Methodology, Section 3 has the detail about UNL, Section 4 describes Bangla grammar especially verb, verb roots, etc. in detail. In Section 5 and 6 we discuss about the dictionary entries which we have designed and developed to convert Bangla sentence. Finally, Section 7 draws conclusions with some remarks on future works.

## 2. LITERATURE REVIEW

For converting Bangla sentence to UNL expressions firstly, we have gone through Universal Networking Language (UNL) [3, 4, 5, 6, 7, 8] where we have learnt about UNL expression, Relations, Attributes, Universal Words, UNL Knowledge Base, Knowledge Representation in UNL, Logical Expression in UNL, UNL systems and specifications of Enconverter. All these are key factors for preparing Bangla word dictionary, enconversion and deconversion rules in order to convert a natural language sentence (here Bangla sentence) into UNL expressions. Secondly, we have rigorously gone through the Bangla grammar [9, 10, 11], Verb and roots (Vowel ended and Consonant Ended) [9, 10, 12], Morphological Analysis [12, 13, 14], Primary suffixes [9, 10, 14, 15], construction of Bangla sentence [9] based on semantic structure. Using above references we extort ideas about Bangla grammar for morphological and semantic analysis in order to prepare Bangla word dictionary (for verb root, verbal inflections, etc)in the format of UNL provided by the UNL center of the UNDL Foundation.

## 3. UNIVERSAL NETWORKING LANGUAGE

The UNL is an acronym for “Universal Networking Language”. It is a computer language that enables computers to process information and knowledge across the language barriers. It is an artificial language that replicates, in the cyber world, the functions of natural languages in human communication. As a result, it enables people to express all knowledge conveyed by natural languages. It also enables computer to intercommunicate, thus providing people with a linguistic infrastructure for distributing, receiving and understanding multilingual information. [4]

The UNL expresses information or knowledge in the form of semantic network with hyper-node. Different from natural languages, UNL expressions are unambiguous. In the UNL semantic network, nodes represent concepts, and arcs represent relations between concepts. Concepts can be annotated. Since the UNL is a language for computers, it has all the components of a natural language. It is composed of words expressing concepts called “Universal Words”, also referred to as UWs which are inter-linked with other UWs to form sentences. These links, known as “relations”, specify role of each word in a sentence. The subjective meaning intended by the speaker can be expressed through “attributes”.

The “Knowledge Base (UNLKB)” is provided to define semantics of UWs. The UNLKB defines every possible relation between concepts including hierarchical relations and inference mechanism based on inclusion relations between concepts. Thus, the UNLKB provides semantic background of the UNL to make sure the meaning of the UNL expressions is unambiguous.

In the last few years, Machine Translation (MT) techniques have been applied to web environments. The growing amount of available multilingual information on the Internet and the Internet users has led to a justifiable interest on this area. Hundreds of millions of people of almost all levels of education, attitudes and different jobs all over the world use the Internet for different purposes [3], where English is the main language of the Internet. But English is not understandable for most of the people. Interlingua translation programs are needed to develop.

The main goal of the UNL system, which allows users to visualize websites in their native languages, is to provide a common representation for accessing Internet of multilingual websites by the majority of the people over the world. A significant part of the development of any machine translation (MT) system is the creation of lexical resources that the system will use. Dictionaries are of critical importance in MT. They are the largest components of an MT system in terms of the amount of information they hold. Generation of natural language from a machine processable, precise knowledge representation has to grapple with the problem of redundancy and impreciseness inherent in any natural language. In the UNL System [4], natural language analysis has been carried out by the EnConverter (EnCo) [7] associated with Word Dictionary of a native language and language specific analysis rules that converts a native language text into UNL expression and DeConverter (DeCo) [8] also associated with Word Dictionary that converts UNL expression to a variety of native languages using language specific generation rules. Our goal is to include Bangla in this system with less effort and we have been working for the last 4 years to achieve our goal. As a part of this process we are presently working on the process to include Bangla Consonant Ended Verb Root in the dictionary so that any Bangla sentence having consonant ended verb toots can be converted to UNL expression. In the following sections we have described the process more elaborately.

## 4. BANGLA VERB, VERB ROOT, VERB INFLECTIONS

Bangla or Bengali (বাংলা, pronounce as bangla) is an eastern Indo-Aryan language. It is native to the region of eastern South Asia known as Bengal, which comprises present day Bangladsh, the Indian state of West Bengal and parts of the Indian states of Tripura and Asam. It is sixth widely spoken language in the world with more than 230 million speakers. Bengala is written in a script called the Bengali script. Like other Indian languages, the letters in the Bengali script are grouped together based on the way they are pronounced. The first 11 letters are all vowels followed by consonants and finally the semi vowels. The consonants are grouped based on how they are pronounced. First comes the velar consonants, then the palatal, the retroflex, the dental, and the labial consonants.

As in this paper we have concentrated on Consonant ended verb root, lets discuss on bangla verb and related terms before describing our work.

### 2.1 Verb (ক্রিয়া)

Bangla verbs are highly inflected and are regular with only few exceptions. They consist of a stem and an ending; they are traditionally listed in Bangla dictionaries in their "verbal noun" form, which is usually formed by adding –a(আ) to the stem, for instance, রাখা (rakha) = "to put or place". The stem can end in either a vowel or a consonant. Verbs are conjugated for tense and person by changing the endings, which are largely the same for all verbs. However, the stem vowel can often change as part of the phenomenon known as “vewel harmony”, whereby one vowel can be influenced by other vowels in the word to sound more harmonious. An example would be the verb "to write", with stem lekh-: তোমরা লিখ (tomra likho) meaning “you (pl.) write” but আমরা লিখি (amra likhi) meaning “we write”.

Bangla language has more than 30000 [9] verbs. Diversity of verb morphology in Bangla is very significant. For example, if we consider “লিখ্” (likh means write) as a root word then after adding verbal inflexion “ইতেছি” (itechhi), we get a word “লিখিতেছি” (likhitechhi means am writing) which means a work is being doing in present (for first person). Similarly, after adding inflexion “ইতেছিলাম” (itechhilam) we get the word “লিখিতেছিলাম”(likhitechhilam means was writing) which means a work was being done in past. Here, one word represents present continuous tense of the root word “লিখ্” (likh) and another represents past continuous tense. Therefore, by morphological analysis we get the grammatical attributes of the main word and other attributes. For this reason we have applied morphological analysis for different persons with different transformations to find out the actual meaning of the word. Morphological analysis of Bangla

verbs has been considered in different works [1, 2]. We show some data for root verbs যা (go) shown in Table 1.

| Person/Tense | Verb as appears in a sentence | Inflectional Suffix |
|---|---|---|
| First | | |
| Present | যাই (jai) | ই ( i ) |
| Present Continuous | যাচ্ছি (jachhi) | চ্ছি (chhi) |
| Present Perfect | গিয়েছি (giechhi) | এছি (chhi) |
| Past | গেলাম (gelam) | লাম ( lam ) |
| Past Continuous | যাচ্ছিলাম (jachhilam) | চ্ছিলাম (chhilam) |
| Past Perfect | গিয়েছিলাম (giechhilam) | এছিলাম (echhilam) |
| Past Habitual | যেতাম (jetam) | এতাম (etam) |
| Future | যাবো (jabo) | বো (bo) |
| Second | | |
| Present | যান (jan) | ন (n) |
| Present Continuous | যাচ্ছেন (jachhen) | চ্ছেন (chhen) |
| Present Perfect | গিয়েছেন (giechhen) | এছেন (echhen) |
| Past | গেলেন (gelen) | এলেন (elen) |
| Past Continuous | যাচ্ছিলেন (jachhilen) | চ্ছিলেন (chhlen) |
| Past Perfect | গিয়েছিলেন (giechhilen) | এছিলেন (echhilen) |
| Past Habitual | যেতেন (jeten) | এতেন (eten) |
| Future | যাবেন (jaben) | বেন (ben) |
| Third | | |
| Present | যায় (jae) | য় (e) |
| Present Continuous | যাচ্ছে (jachhe) | চ্ছে (chhe) |
| Present Perfect | গিয়েছে (giechhe) | এছে (echhe) |
| Past | গেলো (gelo) | লো (lo) |
| Past Continuous | যাচ্ছিলো (jachhilo) | চ্ছিলো (chhilo) |
| Past Perfect | গিয়েছিলো (giechhilo) | এছিলো (echhilo) |
| Past Habitual | যেতো (jeto) | তো (to) |
| Future | যাবে (jabe) | বে (be) |

**TABLE 1:** Morphology of root verb যা (go)

As verbs come from roots and verbal inflexios, for appropriate morphological analysis, we have divide Bangla verb roots into two categories.

- **Vowel Ended Roots:** In Bangla there are around 25 vowel ended roots e.g পা (pa),খা (kha),গা (ga), চা (cha), ছা (chha), নি (ni), দি (di), যা (ja), ছুঁ (chhu), থু (thu), শু (shu), ধু (ddhu),

ন (n), দু (dhu), নু (nu), রু (ru), হ (h), ধা (dha), না (na), বা (ba), ক (ko), ব (bo), র (ro) and স (sho).

- **Consonant Ended Roots**: There are around around 1500 consonant ended roots in Bangla Language. For examples, কর্ (kor), খেল্ (khel), গড় (gor), ঘস্ (gosh), বখ্ (bokh), কহ্ (koh), গিল্ (gil), পিষ্ (pish), শিখ্ (shikh), লিখ্ (likh), ধর্ (dhor), উঠ্ (ut), বুজ্ (buj), ভুল্ (vul) etc. are consonant ended roots.

## 5. Our Proposed Template for Consonant Ended Verb Roots

For appropriate morphological analysis and designing template of verb roots, verb roots have been divided into two broad categories according to tenses and persons namely Vowel Ended Group (VEG) and Consonant Ended Group (CEG). Each of them again divided into sub-groups. Categorization of Consonant Ended Groups are shown in the following tables. (for first person only)

| | | **Consonant Ended Roots** |
|---|---|---|
| | Tenses | কম্,কর্,কশ্,কষ্,খেঁচ্,খেপ্,খেল্,গড়্,গল্,ঘট্,ঘষ্,ঘেঁষ্,<br>চট্,চড়্,চর্,চল্,চষ্,চেত্,ছড়্,ছল্,ছেঁক্,ছেঁচ্,জপ্,জম্,জর্,<br>জ্বল্,ঝর্,টক্,টল্,ঠক্,ঠেক্,ঠেল্,ঠেস্,ডল্,ঢল্,দম্,দল্,দেখ্,ধর,<br>ধস্,পড়্,পর্,ফল্,ফেল্,বক্,বখ্,বন্,বল্,বস্,বেচ্,বেড়্,<br>বেল্,ভজ্,ভর্,মজ্,মল্,মেল্,রট্,রস্,রোপ্,<br>লড়্,লেপ্,সঁপ,সর্,সেঁক,সেক,সেচ,হট্, হর্,হের্,হেল্,হ্রেস,খস্,ঘোষ্ |
| Present | Present Indef | ই |
| | Present Cont | ছি |
| | Present Perfect | এছি |
| | Imperative | * |
| Past | Past Indef | লাম |
| | Past Habitual | তাম |
| | Past Cont. | ছিলাম |
| | Past Perfect | এছিলাম |
| Future | Future Indef. | বো, ব |
| | Imperative | * |
| | | Group CEG1 |

**TABLE 2**: Variations of Consonant Ended Roots and their Verbal Inflexions of CEG1 for First Person

| | | Consonant Ended Roots | |
|---|---|---|---|
| | **Tenses** | কহ্,দহ্,বহ্,রহ্,সহ | **নহ্** |
| Present | Present Indef | ই | ই |
| | Present Cont | ইতেছি | * |
| | Present Perfect | ইয়াছি | * |
| | Imperative | * | * |
| Past | Past Indefinite | ইলাম | * |
| | Past Habitual | ইতাম | * |
| | Past Continuous | ইতেছিলাম | * |
| | Past Perfect | ইয়াছিলাম | * |
| Future | Future Indefinite | ইব, ইবো | * |
| | Imperative | * | * |
| | | **Group CEG2** | **Group CEG3** |

**TABLE 3** Variations of Consonant Ended Roots and their Verbal Inflexions of CEG2 and CGE3 for First Person

| | | Consonant Ended Roots |
|---|---|---|
| | **Tenses** | আঁক্,আঁচ্,আঁট্,আস্,কাঁক,কাঁচ,কাঁড়,কাঁদ্,কাঁপ্,কাচ্,কাট্,কাড়,কান্দ্,কাশ্,খাট্,খাপ্,গাঁজ,গাঁখ্,গাড়গাদ্,গাব্,গাল্,ঘাঁট্, ঘাম্,চাঁচ্,চাখ্,চাট্,চাপ্,চাল্,ছাঁক্,ছাঁট্,ছাঁদ্,ছাড়্,ছান্,জাঁক্,জাগ্,জার্,জ্বাল, ঝাঁক্,ঝাড়্, ঝাঁপ্,ঝাল্, টান্, ঠার্, ঠাস্, ডাক্,ঢাক্,ঢাল্,তাত্,তাপ্,থাক্,থাম্,দাগ্,দাব্,ধাঁদ,ধাঁধ্,ধার্,নাচ্,নাড়্, নাম্ ,পাক্,পাড়্,পাত্,পার্,ফাঁদ্ ,ফাঁপ্ ,ফাঁস্,ফাট্, ফাড়্, বাঁক্, বাঁচ্, বাঁট্,বাঁধ্,বাছ্,বাজ্,বাট্,বাড়্, বাস্,ভাঁজ্,ভাগ্, ভাঙ্,ভাঙ্গ্ ,ভাজ্, ভান্, ভাব্, ভালবাস্, ভাস্, মাখ্, মাগ্, মাঙ্গ্, মাজ্, মাড়্, মাত্,মান্,মাপ্,মার্,যাচ্,রাঁধ,রাখ্,রাগ্, লাগ্,লাদ্, শান্, শাস্, সাজ্, সাঁট্,সাধ্,সান্, সার্, হাঁক্,হাঁচ্,হাজ্,হাট্,হান্,হার্,হাস্ |
| Present | Pre Ind | ই |
| | Present Cont | ছি |
| | Prese Perf | আঁক্>এঁক এছি |
| | Imperative | * |
| Past | Past Indef | লাম |
| | Past Habit | তাম |
| | Past Conti | ছিলাম |
| | Past Perfect | আঁক্>এঁক এছিলাম |
| Future | Future Indef | ব |
| | Imperative | * |
| | | **Group CEG4** |

**TABLE 4:** Variations of Consonant Ended Roots and their Verbal Inflexions of CEG4 for First Person

| | | Consonant Ended Roots | |
|---|---|---|---|
| | **Tenses** | কিন্,খিঁচ্,গিল্,ঘির্,চিন্,চির্,ছিঁড়্,জিত্,জিন্,টিক্,টিপ্,নিব্, পিজ্,পিট্,পিষ্,ফির্,বিধ্,ভিজ্,ভিড়্,মিট্,মিল্,মিশ্,লিখ্,শিখ্,সিচ্ | **আছ্** |
| Present | Pres. Indef. | ই | ই |
| | Pres. Cont | ছি | * |
| | Pres. Perf. | এছি | * |
| | Imperative | * | * |
| Past | Past Indef | লাম | * |
| | Past Habit. | তাম | * |
| | Past Cont. | ছিলাম | * |
| | Past Perf. | এছিলাম | * |
| Future | Fut Indef | ব | * |
| | Imperative | * | * |
| | | **Group CEG5** | **Group CEG6** |

**Table 5:** Variations of Consonant Ended Roots and their Verbal Inflexions of CEG5 and CEG6 for First Person

| | | Consonant Ended Roots |
|---|---|---|
| | **Tenses** | উঠ্,উড়্,উব্,কুঁদ্,কুদ্,কুট্,কুর্,খুঁজ্,খুঁট্,খুঁড়্,খুদ্,খুল্,গুঁজ্,গুণ্,গুল্,ঘুঁট্,ঘুচ্, ঘুর্, চুক্, চুন্,চুষ্,ছুঁড়্, ছুট্,ছুড়্,ছুল্,জুট্,জুড়্,ঝুল্,টুক্, টুট্,ঠুক্,ঠুস্,ডুব্, ঢুক্, ঢুঁড়্, ঢুল্, তুল্, তুষ্,দুল্,দুষ্,ধুঁক্,ধুন্,পুঁছ্,পুঁত্,পুছ্,পুড়্,পুর্,পুষ্,ফুঁক্, ফুঁড়্, ফুট্, ফুল্, বুঁজ্, বুজ্,বুঝ্,বুন্,বুল্,ভুগ্,ভুল্,মুছ্,মুড়্,মুত ,মুদ্,রুখ্, রুধ্,লুট্, লুফ্, শুঁক্, শুধ্, শুন্,শুষ্ |
| Present | Present Indef. | ই |
| | Present Cont | ছি |
| | Present Perfect | এছি |
| | Imperative | * |
| Past | Past Indef | লাম |
| | Past Habitual | তাম |
| | Past Cont. | ছিলাম |
| | Past Perfect | এছিলাম |
| Future | Future Indef. | ব |
| | Imperative | * |
| | | **Group CEG7** |

**Table 6:** Variations of Consonant Ended Roots and their Verbal Inflexions of CEG7 for First Person

We have also developed such template for second and third person also.

## 6. OUR PROPOSED TEMPLATE FOR DICTIONARY ENTRIES

The template that has been developed in this paper for Bangla verb roots is depicted bellow:

**[HW] {} "UW(icl/iof…>concept1>concept2…, REL1>…,REL2>…," (ROOT, VEND/CEND [,ALT/ ALT1/ALT2..] VEGn/CEGn, #REL1, #REL2, … <FLG, FRE, PRI>**

where,

- HW← Head Word (Bangla Word; in this case it is Bangla root);
- UW← Universal Word (English word from knowledge base);
- icl/iof/… means inclusion/instance of …to represent the concept of universal word
- REL1/REL2.., indicates the related relations regarding the corresponding word.
- ROOT ← It is an attribute for Bangla roots. This attribute is immutable for all Bangla roots.
- CEND and VEND are the attributes for consonant ended and vowel ended roots respectively as every root is ended either with consonant or vowel;
- VEGn ← means attribute for the group number of vowel ended roots
- CEGn ← means attribute for the group number of consonant ended roots
- ALT, ALT1, ALT2 etc. are the attributes for the first, second and third alternatives of the vowel or consonant ended roots respectively. If the root is default, then no alternative is used.
- #REF1, #REF2 etc. are the possible corresponding relations regarding the root word.

Here, attributes say, ROOT, CEND/VEND are fixed for all Bangla roots whereas ALT or ALT1 or ALT2 etc. does not necessary for all roots, they are used only for alternative roots.

In the following examples we are constructing the dictionary entries for some sample Bangla roots using our designed template:

[কর্,]{}"do(icl>do, agt>thing, obj>process)" (ROOT, CEND, CEG1, #OBJ, #AGT,#PLC)<B,0,0>
[আঁক্]{}"draw(icl>get>do,equ>reap,src>thing,agt>thing,obj>thing)"(ROOT,CEND, CEG4,#AGT,#OBJ,#PLC) <B,0,0>
[কিন্]{}"buy(icl>get>do,cob>thing,src>thing,agt>person,obj>thing)"(ROOT, CEND,CEG5,#AGT,#OBJ,#PLC)<B,0,0>

where, ROOT denotes Bangla root, CEND for consonant ended root, CEG for consonant ended group. #AGT, #OBJ and #PLC indicate that concerned headwords can be used to make agent (agt), object (obj) and place (plc) relation respectively. Similarly, other entries have been developed according to the format discussed above.

## 7. CONCLUSION AND FUTURE WORK

We have shown dictionary entries for Bangla Consonant Ended Roots that are useful for conversion of Bangla sentences to UNL expression. So with this we have developed dictionary entries for both i.e. vowel ended and consonant ended verb roots. Now in future we'll work on developing rules for these and then we'll use the en- converter to see the result.

## 8. REFERENCES


1. Md. Ershadul H. Choudhury, Md. Nawab Yousuf Ali, Mohammad Zakir Hossain Sarker, Ahsan Razib. "Bridging Bangla to Universal Networking Language- A Human Language Neutral Meta-Language", In proceedings International Conference on Computer, and Information Technology (ICCIT), Dhaka, 2005 pp. 104-109
2. Md. Nawab Yousuf Ali, Mohammad Zakir Hossain Sarker , Ghulam Farooque Ahmed , Jugal Krishna Das."Conversion of Bangla Sentence into Universal Networking Language Expression", International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011, pp. 64-73
3. H. Uchida, M Zhu. T.G. Della Senta. Universal Networking Language, 2005/6-UNDL Foundation, International Environment House.
4. H. Uchida, M. Zhu. The Universal Networking Language (UNL) Specification Version 3.0 Edition 3 , Technical Report, UNU, Tokyo:, 2005/6-UNDL Foundation, International Environment House, 2004
5. Enconverter Specification Version 3.3, UNU Centre, Tokyo 150-8304, Japan 2002
6. http://www.undl.org last accessed on 29 July 2012
7. EnConverter Specification, Version 3.3, UNL Center/UNDL Foundation, Tokyo 150-8304, Japan 2002
8. DeConverter Specification, Version 2.7, UNL Center, UNDL Foundation, Tokyo 150-8304, Japan 2002
9. D.M. Shahidullah. Bangla Baykaron, Dhaka: Ahmed Mahmudul Haque of Mowla Brothers prokashani, 2003
10. D. C. Shuniti Kumar. Bhasha-Prakash Bangala Vyakaran, Calcutta : Rupa and Company Prokashoni, July 1999, pp.170-175
11. Humayun Azad. Bakkototyo - Second edition, Dhaka: Bangla Academy Publishers, 1994
12. D. S. Rameswar. Shadharan Vasha Biggan and Bangla Vasha, Pustok Biponi Prokashoni, November 1996, pp.358-377
13. M.N.Y. Ali, J.K. Das, S. M. Abdullah Al Mamun, M. E. H. Choudhury. "Specific Features of a Converter of Web Documents from Bengali to Universal Networking Language", International Conference on Computer and Communication Engineering 2008 (ICCCE'08), Kuala Lumpur, Malaysia. pp. 726-731
14. M.N.Y. Ali, J.K. Das, S.M. Abdullah Al Mamun, A. M. Nurannabi. "Morphological Analysis of Bangla words for Universal Networking Language", International Conference on Digital Information Management, icdim, 2008, London, England, pp. 532-537
15. M.N.Y.Ali, A. M. Nurannabi, G. F. Ahmed, J.K. Das. "Conversion of Bangla Sentence for Universal Networking Language", International Conference on Computer and Information Technology (ICCIT), Dhaka, 2010 pp. 108-113

# Hybrid Phonemic and Graphemic Modeling for Arabic Speech Recognition

**Mohamed Elmahdy** *mohamed.elmahdy@qu.edu.qa*
*Qatar University*
*Qatar*

**Mark Hasegawa-Johnson** *jhasegaw@illinois.edu*
*University of Illinois*
*USA*

**Eiman Mustafawi** *eimanmust@qu.edu.qa*
*Qatar University*
*Qatar*

**Abstract**

In this research, we propose a hybrid approach for acoustic and pronunciation modeling for Arabic speech recognition. The hybrid approach benefits from both vocalized and non-vocalized Arabic resources, based on the fact that the amount of non-vocalized resources is always higher than vocalized resources. Two speech recognition baseline systems were built: phonemic and graphemic. The two baseline acoustic models were fused together after two independent trainings to create a hybrid acoustic model. Pronunciation modeling was also hybrid by generating graphemic pronunciation variants as well as phonemic variants. Different techniques are proposed for pronunciation modeling to reduce model complexity. Experiments were conducted on large vocabulary news broadcast speech domain. The proposed hybrid approach has shown a relative reduction in WER of 8.8% to 12.6% based on pronunciation modeling settings and the supervision in the baseline systems.

**Keywords:** Arabic, Acoustic modeling, Pronunciation modeling, Speech recognition.

## 1. INTRODUCTION

Arabic is a morphologically very rich language that is inflected by gender, definiteness, tense, number, case, humanness, etc. Due to Arabic morphological complexity, a simple lookup table for phonetic transcription -essential for acoustic and pronunciation modeling- is not appropriate because of the high out-of-vocabulary (OOV) rate. For instance, in Arabic, a lexicon of 65K words in the domain of news broadcast leads to an OOV rate in the order of 5% whilst in English, it leads to an OOV rate of less than 1%.

Furthermore, Arabic is usually written without diacritic marks. Text resources without diacritics are known as non-vocalized (or non-diacritized). These diacritics are essential to estimate short vowels, nunation, gemination, and silent letters. The absence of diacritic marks leads to a high degree of ambiguity in pronunciation and meaning [10, 13].

In order to train a phoneme-based acoustic model for Arabic, the training speech corpus should be provided with fully vocalized transcriptions. Then, the mapping from vocalized text to phonetic transcription is almost a one-to-one mapping [10]. State of the art techniques for Arabic vocalization are usually done in several phases. In one phase, orthographic transcriptions are manually written without diacritics. Afterwards, statistical techniques are applied to restore missing diacritic marks. This process is known as “automatic diacritization”. Automatic diacritization techniques can results in diacritization WER of 15%-25% as reported in [10, 12, 16].

In order to avoid automatic or manual diacritization, graphemic acoustic modeling was proposed for Modern Standard Arabic (MSA) in [8] where the phonetic transcription is approximated to be the sequence of word letters while ignoring short vowels. Missing short vowels are assumed to be implicitly modeled in the acoustic mode. It could be noticed that graphemic systems work with an acceptable recognition rate. However the performance is still below the accuracy of phonemic models. Graphemic modeling has an advantage of the straightforward pronunciation modeling approximation, as pronunciation is directly estimated by splitting the word into letters.

Large language models are usually trained with large amounts of non-vocalized text resource. In order to use large language models along with phonemic acoustic model, the pronunciation model should explicitly provide the possible phonemic pronunciations. A morphological analyzer such as the Buckwalter Morphological Analyzer [18] can be used to generate all possible diacritization forms for a given word. This approach was widely used in many Arabic speech recognition systems as in the GALE project and other systems [1, 7, 9]. Actually, this technique results in multiple pronunciation variants for each word. The problem with this approach is that Arabic has a high homograph rate. Hence, the same word has a large number of possible pronunciation variants. In other words, the morphological analyzer provides much more variants than required, and most of them are legacy non-common pronunciations. This large number of variants makes the distance between the different pronunciations becomes very small and hence results in more recognition errors. Another problem with this approach is that pronunciation variants cannot be estimated for words that are not morphologically parsable (e.g. named entities and dialectal words).

In this research, we propose a hybrid modeling approach that can benefit simultaneously from both the grapheme-based and the phoneme-based techniques. Our assumption is that for a relatively small vocalized text corpus, high frequency words always exist (e.g. في , من , على , etc). On the other hand, for larger non-vocalized corpora, we have better lexical coverage for low frequency words. By combining a phonemic acoustic model along with a graphemic model, we can benefit from little amounts of vocalized text for accurate pronunciation modeling of high frequency words. Moreover, for low frequency words, graphemic modeling is still possible and the pronunciation model will not fail.

## 2. SPEECH CORPORA

Three speech corpora have been chosen in our work. All of them are from the domain of news broadcast. Two corpora were sourced from the European Language Resources Association (ELRA) [6] and the third one was sourced from the Linguistic Data Consortium (LDC) [11]. All resources were recorded in linear PCM format, 16 kHz, and 16 bit. The ELRA speech resources were:

- The NEMLAR Broadcast News Speech Corpus: consists of ~40 hours from different radio stations: Medi1, Radio Orient, Radio Monte Carlo, and Radio Television Maroc. The recordings were carried out at three different periods between 30 June 2002 and 18 July 2005. The corpus is provided with fully vocalized transcriptions [17].
- The NetDC Arabic BNSC (Broadcast News Speech Corpus): contains ~22.5 hours of broadcast news speech recorded from Radio Orient during a three month period between November 2001 and January 2002. The orthographic transcriptions are fully vocalized with the same guidelines as the Nemlar corpus. Detailed composition of the ELRA databases is shown in Table 1.

The LDC resource was the Arabic Broadcast News Speech (ABNS) corpus [11]. The corpus consists of ~10 hours recorded from the Voice of America satellite radio news broadcasts. The recordings were made at time of transmission between June 2000 and January 2001. The orthographic transcription provided with this corpus is partially vocalized.

The two ELRA resources have been taken as the training set (~62 hours) and the LDC corpus has been taken as the testing set (~10 hours) as shown in Table 1. This way we can guarantee complete independence between the training and the testing sets including recording setups, speakers, channel noise, time span, etc.

| **Training set** | | |
|---|---|---|
| **Corpus** | **Source** | **Duration (hours)** |
| NetDC | Radio Orient | 22.5 |
| Nemlar | Radio Orient<br>Medi1<br>Radio Monte Carlo<br>Radio Tele. Maroc | 12.1<br>9.5<br>9.0<br>9.3 |
| **Testing set** | | |
| **Corpus** | **Source** | **Duration (hours)** |
| ABNS | Voice of America | 10.0 |

**TABLE 1:** Composition of the Arabic speech broadcast news resources.

## 3. LANGUAGE MODELING

Two language models have been trained: a small language model (Small-LM-380K) and a large language (Large-LM-800M). The two models are backoff tri-gram models with Kneser-Ney smoothing. The Small-LM-380K model has been trained with the transcriptions of the speech training set (~380K words) that consists of 43K unique words after eliminating all diacritic marks. The evaluation of the Small-LM-380K model against the transcriptions of the speech testing set resulted in an OOV rate of 10.1%, tri-grams hit of 19.6%, and perplexity of 767.5 (entropy of 9.6 bits) as shown in Table 2.

The Large-LM-800M has been trained with the LDC Arabic Gigaword fourth edition corpus [15] that consists of ~800M words. Vocabulary was chosen to be the top 250K unique words. The evaluation of the Large-LM-800M model against the transcriptions of the speech testing set resulted in an OOV rate of 3.1%, tri-grams hit of 41.8%, and perplexity of 464.6 (entropy of 8.9 bits) as shown in Table 2.

Language modeling in this research was carried out using the CMU-Cambridge Statistical Language Modeling Toolkit [2, 14].

| **Language Model** | Small-LM-380K | Large-LM-800M |
|---|---|---|
| **Training words** | 380K | 800M |
| **Vocabulary** | 43K | Top 250K |
| **OOV** | **10.1%** | **3.1%** |
| **Perplexity** | 767.5 | 464.6 |
| **Tri-grams hit** | 19.6% | 41.8% |

**TABLE 2:** Language models properties and evaluation against the transcriptions of the testing set.

## 4. SYSTEM DESCRIPTION

Our system is a GMM-HMM architecture based on the CMU Sphinx engine [3, 4]. Acoustic models are all fully continuous density context-dependent tri-phones with 3 states per HMM trained with MLE. The feature vector consists of the standard 39 MFCC coefficients. During acoustic model training, linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT) were applied to reduce dimensionality to 29 dimensions. This was found to improve accuracy as well as recognition speed. Decoding is performed in multi-pass, a fast

forward Viterbi search using a lexical tree, followed by a flat-lexicon search and a best-path search over the resulting word lattice.

## 5. PHONEMIC BASELINE SYSTEM

### 5.1 Phonemic Acoustic Modeling

The 62 hours training set was used to train the phonemic acoustic model. A grapheme-to-phoneme module was developed to convert the vocalized transcriptions to phonetic ones. The phoneme set consists of 28 consonants, 3 short vowels, and 3 long vowels. Diphthongs were treated as two consecutive phonemes (a vowel followed by a semi-vowel). The acoustic model consists of both context-independent (CI) and context-dependent (CD) phones. During decoding, CI models were used to compute the likelihood for tri-phones that have never been seen in the training set. The CI models consist of 102 states with 32 Gaussians each. The total number of CD tied-states is 3000 with 32 Gaussians each.

### 5.2 Phonemic Pronunciation Modeling

Phonemic pronunciation modeling is done through a lookup table lexicon, where each entry word is associated with one or more pronunciation variants. The lexicon was built using the phonetic transcription of the training data set (380K words), resulting in 43K unique words with an average of ~1.6 variants per word. Each word is also associated with a rank based on its frequency in the vocalized text resource we have used.

## 6. Graphemic Baseline System

### 6.1 Graphemic Acoustic Modeling

All diacritic marks have been removed from the transcriptions of the training set. A graphemic acoustic model was trained by approximating the pronunciation to be the word letters rather than the actual pronunciation. Each letter was mapped to a unique model resulting in a total number of 36 base units (letters in the Arabic alphabet). The graphemic acoustic model consists of 108 CI states and 3000 CD tied-states with 32 Gaussians each.

### 6.2 Graphemic Pronunciation Modeling

In this case, pronunciation modeling is a straightforward process. For any given word, pronunciation modeling is done by splitting the word into letters. Each word is associated with only one graphemic pronunciation. The major advantage of this modeling technique is the ability to generate a model for any word. However, it is still just an approximation to overcome the problem of out of lexicon words in phonemic modeling.

## 7. Hybrid System

### 7.1 Hybrid Acoustic Modeling

Hybrid acoustic modeling is performed by combining the phonemic and the graphemic models into one hybrid model after two independent trainings. This results in having all the phonemic and graphemic HMMs into the same model. The final model consists of 70 base units (34 phonemic and 36 graphemic) with 210 CI states. The total number of CD tied-states is 6000 (3000 phonemic and 3000 graphemic).

One limitation of hybrid acoustic modeling is that the acoustic model does not contain cross tri-phones between graphemic and phonemic units. These types of tri-phones can appear between two words, one with grapheme-based pronunciation and the other one with phoneme-based pronunciation. In this case, the decoder in our system backs off to CI units to compute acoustic likelihood.

### 7.2 Hybrid Pronunciation Modeling

In hybrid modeling, pronunciation is estimated from a phonemic lexicon in conjunction with the previously discussed graphemic approach. The variants in this case can be phonemic and/or graphemic. The decoder selects the appropriate acoustic phone models (either phonemic or graphemic), based on the pronunciation(s) generated by the pronunciation model. The phonemic lexicon is the same 43K lexicon used in the phonemic baseline. For any given non-vocalized word, three different hybrid pronunciation modeling techniques are proposed: Hybrid-Or, Hybrid-And, and Hybrid-Top(n).

#### 7.2.1 Hybrid-Or

In the Hybrid-Or approach, either graphemic modeling or phonemic modeling is applied for any given word. The approach is adopted as follows:

1. Check the existence of the word in the phonemic lexicon.
2. If the word does not exist in the lexicon, only one graphemic variant is generated.
3. If the word exists in the lexicon, all phonemic pronunciation variants associated with that entry word are extracted from the lexicon.

According to our assumption that high frequency words always exist in the lexicon, this means that for a high frequency word like من , only phonemic variants are generated: /m i n/, /m i n a/, and /m a n/.

The drawback of this approach is that for low frequency words that might appear in the lexicon, the generated phonemic variant cannot model all possible variations. That is because low frequency words have always a low rank in the lexicon lacking the coverage for all possible variants.

#### 7.2.2 Hybrid-And

In the Hybrid-And approach, a graphemic pronunciation is always generated for any given word in addition to the existing phonemic pronunciations in the lexicon as follows:

1. Check the existence of the word in the phonemic lexicon.
2. If the word does not exist in the lexicon, only one graphemic variant is generated.
3. If the word exists in the lexicon, one graphemic variant is generated in addition to all existing variants in the lexicon.

In this approach, we are trying to compensate low ranked words in the lexicon, with one generic graphemic variant to model the missing variants. The drawback is that we also generate a redundant graphemic variant for high frequency words as well, and this might decrease recognition rate. For instance, the word من will have one redundant graphemic variant /م ن/ and the phonemic variants: /m i n/, /m i n a/, and /m a n/.

#### 7.2.3 Hybrid-Top(n)

The Hybrid-Top(n) approach is a mixture of Hybrid-Or and Hybrid-And. For n=N, pronunciation modeling is performed as follows:

1. Check the existence of the word in the phonemic lexicon.
2. If the word does not exist in the lexicon, only one graphemic variant is generated.
3. If the word exists in the lexicon, check the word's rank in the phonemic lexicon.
    - If the word exists among the Top(N) high frequently used words, only the phonemic variants associated with that entry word are generated.
    - If the word's rank is below the Top(N) words, one graphemic variant is generated in addition to all existing variants in the lexicon.

In the Hybrid-Top(n) approach, we are trying to keep only phonemic pronunciations for high frequency words. On the other hand, for low frequency words, a generic graphemic model is added to compensate missing variants.

## 8. Recognition Results

### 8.1 Phonemic Modeling Results

Performance was evaluated against the 10 hours testing set. The phonemic acoustic model along with the Small-LM-380K language model resulted in a WER of 47.8% as shown in Table 3. A significant percentage of the errors was due to the high OOV rate. Moreover, phonemic modeling for low frequency was less accurate than high frequency words. The large-LM-800K resulted in a WER of 41.1%, as shown in Table 4, with a relative reduction in WER of 14.0% compared to the case of Small-LM-380K. The relative reduction in WER shows only the effect of a larger language model. In the case of the large language model, pronunciation modeling suffers from the inability of generating the appropriate pronunciation for words that do not exist in the lexicon.

### 8.2 Graphemic Modeling Results

The graphemic acoustic model along with the Small-LM-380K language model resulted in a WER of 53.4%, as shown in Table 3, with a relative increase in WER of 11.7% compared to phonemic modeling. This relative increase in WER shows the difference in performance between the phonemic and the graphemic approach when each word in the language model has a phonemic pronunciation.

The Large-LM-800K resulted in a WER of 42.1%, as shown in Table 4, with a relative increase of 2.4% compared to the case of phonemic modeling. The small relative difference is mainly interpreted because of the lack in pronunciation modeling for the high number of words in the Large-LM-800K model.

| Acoustic model | WER | Relative |
|---|---|---|
| Phonemic AM | 47.8% | - |
| Graphemic AM | 53.4% | +11.7% |

**TABLE 3:** WERs on the 10 hours testing set using the Small-LM-380K language model and the conventional acoustic modeling techniques: phonemic and graphemic.

### 8.3 Hybrid Modeling Results

Hybrid modeling was first tested with the Small-LM-380K language model. However, no improvement was observed compared to the phonemic baseline system. That was expected since the entire vocabulary of the Small-LM-380K model already exist in the phonemic lexicon.

On the other hand, hybrid modeling along with the Large-LM-800M language model resulted in significant accuracy improvement compared to both the phonemic and the graphemic baselines. The Large-LM-800M along with the hybrid acoustic model were used in decoding the testing set. The three hybrid pronunciation modeling approaches have been evaluated as follows:

#### 8.3.1 Hybrid-Or

In the Hybrid-Or approach, the absolute WER was 37.5% outperforming the phonemic and the graphemic baseline systems by 8.8% and 10.9% relative reduction in WER respectively as shown Table 4.

#### 8.3.2 Hybrid-And

In Hybrid-And settings, the absolute WER was reduced to 36.9% absolute, outperforming the phonemic and the graphemic approaches by 10.2% and 12.4% relative reduction. The improvement in Hybrid-And compared to Hybrid-Or was mainly interpreted as the lexicon does

not have enough variants for low frequency words. That is why by generating a graphemic variant along with available phonemic variants, missing variants can be modeled by the generic graphemic model.

### 8.3.3 Hybrid-Top(N)

The Hybrid-Top(n) approach was evaluated by taking n=100 (i.e. the top 100 most frequently used words), the WER was slightly improved achieving 36.8% absolute WER, outperforming the phonemic and the graphemic approaches by 10.5% and 12.6% relative reduction in WER respectively.

The slight improvement is interpreted since high frequency words in the lexicon are already associated with almost all possible pronunciation variants, and by adding an extra graphemic variant, the distance between them become smaller and more recognition errors are expected. That is why eliminating graphemic variants from the top words may slightly improve recognition accuracy.

Actually, the slight accuracy improvement of Hybrid-Top(n) compared to Hybrid-And is not significant. Thus, accuracy wise, Hybrid-And and Hybrid-Top(n) are in fact equivalent. However, the advantage of the Hybrid-Top(n) is that it can reduce system complexity by eliminating the redundant graphemic variant associated with the top high frequency words, and hence can improve real time factor.

| Approach | WER | Relative to Phonemic | Relative to Graphemic |
|---|---|---|---|
| Phonemic | 41.1% | - | -2.4% |
| Graphemic | 42.1% | +2.4% | - |
| Hybrid-Or | 37.5% | -8.8% | -10.9% |
| Hybrid-And | 36.9% | -10.2% | -12.4% |
| Hybrid-Top(n), n=100 | 36.8% | -10.5% | -12.6% |

**TABLE 4:** WERs on the 10 hours testing set using the Large-LM-800M language model and the different acoustic and pronunciation modeling techniques.

## 9. Conclusions and Future Work

Arabic is a morphologically very rich language. This morphological complexity results in high OOV rate compared to other languages like English. In large vocabulary speech recognition, the high OOV rate can significantly reduce speech recognition accuracy due to the limitation of pronunciation modeling.

In this paper, we have proposed a hybrid approach for Arabic large vocabulary speech recognition. The proposed approach benefits from both phonemic and graphemic modeling techniques where two acoustic models are fused together after two independent trainings.

First, two baseline systems were built: phonemic and graphemic. With the Small-LM-380K language model where all words in the vocabulary are associated with phonemic pronunciation, the phonemic baseline has outperformed the graphemic baseline by -11.7% relative decrease in WER. With the Large-LM-800M language model, the gap was decreased where the phonemic system has outperformed the graphemic system by -2.4% relative reduction in WER.

In order to create the hybrid acoustic model, the phonemic and the graphemic models were combined together. Three different approaches have been proposed for hybrid pronunciation modeling: Hybrid-Or, Hybrid-And, and Hybrid-Top(n).

The Hybrid-Or technique has resulted in 37.5% absolute WER outperforming the phonemic and the graphemic baselines by -8.8% and -10.9% relative reduction in WER respectively. The hybrid-And technique has resulted in 36.9% absolute WER outperforming the phonemic and the graphemic baselines by -10.2% and -12.4% relative.

Best hybrid modeling approach was found to be the Hybrid-Top(n), where a lexicon-based pronunciation modeling has been used for the top n words and we apply the Hybrid-And approach on the non-top n words. Hybrid-Top(n) results show that hybrid modeling outperforms phonemic and graphemic modeling by -10.5% and -12.6% relative reduction in WER respectively.

In large vocabulary speech domains, acoustic and pronunciation modeling is a common problem among all the Arabic varieties and not only limited to standard Arabic form. Thus, for future work, the proposed approach will be extended and evaluated with the different Arabic colloquials (e.g. Egyptian, Levantine, Gulf, etc.). Moreover, the proposed technique can be also applied on others morphological rich languages like Turkish, Finnish, Korean, etc.

## 10. Acknowledgements

This publication was made possible by a grant from the Qatar National Research Fund under its National Priorities Research Program (NPRP) award number NPRP 09-410-1-069. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Qatar National Research Fund.

We would like also to acknowledge the Linguistic Data Consortium (LDC) and the European Language Resources Association (ELRA) for providing us with the required speech and text resources to conduct this research.

## 11. REFERENCES

[1] A. Messaoudi, L. Lamel, and J. Gauvain, “Transcription of Arabic Broadcast News”. *In International Conference on Spoken Language Processing (INTERSPEECH)*, pp. 1701-1704, 2004.

[2] Carnegie Mellon University-Cambridge, CMU-Cambridge Statistical Language Modeling toolkit, http://www.speech.cs.cmu.edu/SLM/toolkit.html

[3] Carnegie Mellon University Sphinx, Speech Recognition Toolkit, http://cmusphinx. sourceforge.net/

[4] D. Huggins-Daines, M. Kumar, A. Chan, A. W Black, M. Ravishankar, and A I. Rudnicky, “Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices”, *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 185-188, 2006.

[5] D. Vergyri and K. Kirchhoff, “Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition”, *In proceedings of COLING Computational Approaches to Arabic Script-based Language*s, pp. 66-73, 2004.

[6] ELRA: European Language Resources Association, http://www.elra.info/

[7] H. Kuo, S. Chu, B. Kingsbury, G. Saon, H. Soltau, F. Biadsy, “The IBM 2011 GALE Arabic speech transcription system”, *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 272- 277, 2011.

[8] J. Billa, M. Noamany, A. Srivastava, D. Liu, R. Stone, J. Xu, J. Makhoul, and F. Kubala, “Audio indexing of Arabic broadcast news”, *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 5–8, 2002.

[9] L. Lamel, A. Messaoudi, and J. Gauvain, “Automatic Speech-to-Text Transcription in Arabic”, *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4), 2009

[10] M. Elmahdy, R. Gruhn, and W. Minker, “Novel Techniques for Dialectal Arabic Speech Recognition”, *Springer*, 2012.

[11] M. Maamouri, D. Graff, C. Cieri, “Arabic Broadcast News Speech”, *Linguistic Data Consortium(LDC)*, LDC Catalog No.: LDC2006S46, 2006.

[12] N. Habash and O. Rambow, “Arabic Diacritization through Full Morphological Tagging”, *Proceedings of NAACL HLT 2007*, pp. 53-56, 2007.

[13] N. Habash, “Introduction to Arabic Natural Language Processing”, Morgan and Claypool Publishers, 2010.

[14] P. Clarkson, and R. Rosenfeld, “Statistical Language Modeling Using the CMU-Cambridge Toolkit”, *In Proceedings of ISCA Eurospeech*, 1997.

[15] R. Parker, D. Graff, K. Chen, J. Kong, and K. Maeda, “Arabic Gigaword Fourth Edition”, *Linguistic Data Consortium(LDC)*, LDC Catalog No.: LDC2009T30, 2009.

[16] R. Sarikaya, O. Emam, I. Zitouni, and Y. Gao, “Maximum Entropy Modeling for Diacritization of Arabic Text”, *In Proceedings of International Conference on Speech and Language Processing INTERSPEECH*, pp. 145–148, 2006.

[17] The Nemlar project, http://www.nemlar.org/

[18] T. Buckwalter, “Buckwalter Arabic Morphological Analyzer Version 1.0”, *Linguistic Data Consortium(LDC)*, LDC Catalog No.:LDC2002L49, 2002.

# INSTRUCTIONS TO CONTRIBUTORS

Computational linguistics is an interdisciplinary field dealing with the statistical and/or rule-based modeling of natural language from a computational perspective. Today, computational language acquisition stands as one of the most fundamental, beguiling, and surprisingly open questions for computer science. With the aims to provide a scientific forum where computer scientists, experts in artificial intelligence, mathematicians, logicians, cognitive scientists, cognitive psychologists, psycholinguists, anthropologists and neuroscientists can present research studies, International Journal of Computational Linguistics (IJCL) publish papers that describe state of the art techniques, scientific research studies and results in computational linguistics in general but on theoretical linguistics, psycholinguistics, natural language processing, grammatical inference, machine learning and cognitive science computational models of linguistic theorizing: standard and enriched context free models, principles and parameters models, optimality theory and researchers working within the minimalist program, and other approaches. IJCL is a peer review journal and a bi-monthly journal.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, Directory of Open Access Journals (DOAJ), Open J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCL.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Starting with Volume 4, 2013, IJCL aims to appear with more focused issues related to computational linguistics studies. Besides normal publications, IJCL intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

**IJCL List of Topics:**
The realm of International Journal of Computational Linguistics (IJCL) extends, but not limited, to the following:

- Computational Linguistics
- Computational Models
- Computational Theories
- Corpus Linguistics
- Formal Linguistics-Theoretic and Grammar Induction
- Information Retrieval and Extraction
- Language Generation
- Language Learning
- Linguistics Modeling Techniques
- Linguistics Theories
- Machine Translation
- Models of Language Change and its Effect on Lingui
- Models that Address the Acquisition of Word-order
- Models that Combine Linguistics Parsing
- Models that Employ Statistical/probabilistic Gramm
- Models that Employ Techniques from machine learning
- Natural Language Processing
- Quantitative Linguistics
- Speech Analysis/Synthesis
- Speech Recognition/Understanding
- Spoken Dialog Systems
- Web Information

# CALL FOR PAPERS

**Volume:** 4 - **Issue:** 1

**i. Paper Submission:** December 31, 2012 **ii. Author Notification:** January 31, 2013

**iii. Issue Publication:** February 2013

## CONTACT INFORMATION

**Computer Science Journals Sdn BhD**

B-5-8 Plaza Mont Kiara, Mont Kiara

50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6204 5627

Fax: 006 03 6204 5628

Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2012
COMPUTER SCIENCE JOURNALS SDN BHD
B-5-8 PLAZA MONT KIARA
MONT KIARA
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6207 1607
006 03 2782 6991

FAX: 006 03 6207 1697
EMAIL: cscpress@cscjournals.org