1. INTRODUCTION:

In modern Linguistics trend ‘Corpus Analysis’ is a remarkable stream. ‘Corpus Analysis’ is useful in any area of Linguistics like Phonology, Morphology, Syntax, Socio-linguistics, Machine Translation and Computational Linguistics and so on

It focuses on the description of quantitative patterns of Linguistic elements. Mainly it focuses on the Description of Linguistic Performance in a particular language.

Today linguistics believes that what people actually use is the real

language. It reflects on the ideological as well as technological meanderers in

Linguistics. Ideological changes makes from the path of intuition based

rationalistic assumption. The technological change i.e. computers posses and

delivers massive storage facilities and impressive processing. In this project we

deal the corpus analysis for parts of speech tagging rules in technological way i used the PERL programming. Tagging is nothing but labeling. POS Tagging

means we add the Parts of speech category to the word depending upon the context in a sentence. It is also known as Morpho-syntactic Tagging.

In NLP, at the starting stage to any language application, two tools are

necessary.

They are:

1.Morphological Analyzer and

2. POS Tagger.

These two tools are more important for any Natural Language Processing. Another thing here we have to mention is, POS Tagging is also very essential in Machine Translation to understand the Target Language.

For English language there are eight Parts of Speech like

*noun,*

*pronoun,*

*verb,.......*

Further in POS, we classify the categories into sub-categories in deep

analysis of the languages.

1. **TAGS**

With help of Tags, we develop the “Telugu Tag Set”. In Telugu there is

mainly five POSes. They are

1. *Noun*

2. *Pronoun*

3. *Verb*

4. *Adjective*

*5*.Adverbs

We make a sub-classification of these main POS further depending upon

the context.

1. **RESOURCES FOR POS TAGGING**

For POS Tagging it is necessary to use some resources.

They are:

These five are the major resources for POS Tagging.

**A.TEXT & CORPUS**

The Text or Corpus for POS Tagging should be pre-edited and error free.Then only we can use the Text or Corpus for language research. Otherwise there

TEXT

Or

CORPUS

TAG SET

MORPHOSYNTACTIC RULES

TAGGER

MORPHOLOGICAL ANALYSER

**POS** SYMBOLS SET

SYNTACTIC LEVEL

may be some problems in using the Tagger. So we have to prepare the text for

the analysis.

1. **MORPHOLOGICAL ANALYZER**

Morphological analyzer is a basic tool in any NLP Application. Without morphological analysis we can’t go for any computational programming for Indian languages. Some languages particularly Indian languages without morphological analyzer it is very difficult to do any research

1. **TAG SET:**

Tag set is a set, which have different Tags for a particular language. In the first stage manually we will tag different Texts and build a Telugu Tag set and it can cover all the POS in Telugu Language. For that we observe the definitions to ‘corpus-corpora’.

1. **CORPUS-CORPORA**

The word corpus is derived from the Latin word, which means ‘body’.

Corpus is a limited sized body of machine readable texts sampled in order to be

maximally representative of the language variety under consideration. It has the

quality of ‘representativeness.’ There are many definitions to corpus given by

Linguists. Corpus is a collection of naturally occurring texts chosen to

characterize a state or variety of language (Sinclair 1991).

Corpus is collection of running texts which may be spoken,

written or intermediate forms and the sample may be of any

length (Jan Aarts 1991).

Depending upon these types of definitions we can find that there are

mainly four readings to corpus. They are

Sampling & representativeness

Finite size

Corpus Machine readable forms

Standard reference

A collection of huge corpus is considered as corpora. Corpora are the

plural form of Corpus. These corpora also have the same four main headings.

Corpus is different from Text.

1. Texts are representatives of unified communicative events. Corpus is a fragment.

2. A text is read horizontally from left to right paying attention to the boundaries of units. Corpus may be vertical.

3. Text is unique and individualistic, but Corpus is repeated and social.

But there are limitations in corpus study.

They are:

1. Lake of linguistic generativity

2. Technical difficulties

3. Lack of texts from dialogic interactions

4. Lack of information from visual elements.

**TYPES OF CORPORA**

In Corpora the types include texts as well as the combinations. It is very

difficult to design an organized scheme of corpus classification based on the

content. With the features, which are discussed below Corpora are classified into

the following way by usage.

**USES OF CORPORA**

In modern linguistics ‘corpus’ study is very essential part to do any

research activity. Corpus studies show the total structure of a particular language

before our eye. Corpus studies are very useful in speech research, lexical

studies, psycholinguistics and NLP and so on. The main advantages of Corpora

are:

**EASE OF ACCESS USING**

Once the corpus is ready, as in machine-readable form, it is also easy to

access the data. With a program we can quickly retrieve the frequency lists and

indices of various words or other structure of language with in it.

**ENRICHED DATA**

English corpora are available with additional linguistics information like POS

Tagging, parsing. Data retrieval from the corpus can be easier and more specific

than with unannotated data.

**NATURALISTIC DATA**

Corpora should be naturalistic data. Then only it provides the most reliable

source of data on language as it is actually used.

These are the main advantages of corpora. In this project we want to try to

build the Parts of Speech Tagging rules for Telugu based on the Corpus, which is available in Telugu.

1. **Nouns (NN)** :

Noun are often described as referring to persons, places, things, states or Qualities, and the word noun is itself often used as an attributes modifier, as in noun compound.

The different tags in the subclass are depending on the *vibhakti*, the nouns get the number label to main class, i.e., *NN* based on the *karaka* relations. The tag *NN* stands for noun oblique form indicating that the noun is in a position to get attached with the succeeding noun inflection.

SUFFIXES SUFFIXES CALLED

-lo-లొ Locative sig,plu

-lu-లు Nominative plural

-to-తొ Instrumental-societies sig,plu

-ku-కు Dative sig, plu

-ki-కి Dative sig, plu

Some real time examples representation table with suffixes mentioned above:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **ROOT** | **SUFFIXES** | **GRAMMATICAL DESCRIPTION** | **EXAMPLES** |
| A | Doll  /bomma/  /bomma-lu/ | -lu- | NOM,Sig  NOM,plu | Bommalu konnaru |
| B | Doll  /bomma-to/  /bommal-a-to/ | -to-  -a-to- | Inst,sig  Inst,plu | Papa bommal-a-to aduthundi |
| C | Doll  /bomma-lo/  /bommal-(a)-lo/ | -lo-  -(a)-lo- | Locative,sig  Locative,plu | Bomma-lo cotton vunti |
| D | Doll  /bomma-ku/  /bommal-a-ku/ | -ku-  -a-ku- | Dative,sig  Dative,plu | Bommala-ku rangulu veyandi |
| E | House  /inti-ki/  /ill-a-ki/ | -ki-  -a-ki- | Dative,sig  Dative,plu | Illa-ku rungulu veyandi |

1. **Prepositions (Vibhakti – IN):**

A part of speech that indicates the relationship, often spatial, of one word to another. Sometimes prepositions can occur independently. For example, varaku (వరక􀂡). Hence all vibhaktis are labelled as *pp*etc.

**TELUGU POSTPOSITIONS:**

|  |  |
| --- | --- |
| POSTPOSITIONs(Telugu) | ENGLISH Meaning |
| -ki-(కి) | to (somebody or somewhere) |
| -ku-(కు) | to (somewhere) |
| -lo-(లొ) | In |
| -loki-(లోకి) | Into |
| -tho-(తొ) | With |
| -yokka-(యొక్క) | Of |
| -vaipuku-(వైపుకు) | Towards |

1. **Pronouns (sarva nAmAlu - PP) :**

A word that takes the place of a noun.she,he ,it,they,them are examples of pronouns in English.

Personal pronouns in Telugu come in three 'persons': First person, Second person and Third person. These pronouns are different in singular and plural numbers Like nouns, all pronouns form inflections with *vibhaktis*. Accordingly they are named as *PP* etc.

The following are the list of pronoun suffixes and pronoun prefixes:

**Pronoun List in Telugu**:

**Telugu English**

nEnu I

nu-vvu you

mE-mu we(మనం)

ma-naM we (including the person spoken to)

mI-ru you (respectful)

vA-Du he (familiar mode; not respectful)

A adi she (familiar mode; not respectful)

A-ya-na(ఆయన he (respectful / toward a elder person)

A-me she

A-vi-Da she (respectful)

vA-ru he (respectful)

vA-LLu they (plural object)

A a-vi they (non-human objects)

**Pronoun Suffixes List:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Root | Suffixes | Grammatical Description | Example |
| 1  2  3  4 | It  /Wati- ni/  She  /dani-to/  It  /wati-a-lo/  She  /Dani-ki/ | “ni”    “to”  “to”  “ki” | Accusative(pl)    Instrumental-  Sociaive(sg)  Locative(pl) | Wati-ni tisukovellu,  Wati-ni pattuko  Dani-to patu,  Dani-to vellu  Wati-lo-pettu,  Wati-lo unnavi  Dani-ki ,  Dani-ki enni? |

1. **Adjectives (Visheshana-JJ) :**

Adjectives are words that describe or modify another person or thing in the sentence. Special type of adjectives like Verbal adjectives ( *kriya visheshana*) as vjj, Nominal adjectives (*saMjna viseshana*) as ADJ and non infinitive verbal adjectives (*sahAyaka asamapaka kriya*) as ADJ etc.

Adjective ADJ nallani, aMxamEna

**Pronominal ending of Adjectives:**

In Telugu, Adjectives are morphologically marked with pronominal suffixes when they occurred as predicative in equitive or popular sentences, strictly speaking ,this pronominalization is different from that of nouns which occurs with similar inflections.

The pronominal (3rd person) suffixes that accurse with adjectives primarily show agreement in gender number and person. This pronominal endings however, convey different meanings when they are attached with nouns. Consider the fallowing examples.

Ex: xi maMchi kukka Kukka manciXi

1. **Verbs - VB:**

In Telugu,verbs can be identified in two types:

1. Based on the termination of their suffixes from root

2. Based on termination of Tense+PNG(person-number-gender).

It can be explained in below

1. **Based on the termination of their suffixes from root:**

The simple verbs in telugu are in 3 forms

1. tam ending words ex:cheaya+tam (do)

2. dam ending words ex:raya+dam (write)

3. ta endings ex: kottu+ta (strike or beat)

4. yu ending words ex: che+yu (do)

1. **Based on termination of Tense+PNG(person-number-gender):**

In this we obsrved in 3 types:

1. Progressive present tense:

In this the word ending with the roots : nnanu,nnaavu,nnaaru,nnaadi,nnaadu,nnaamu.

Ex: kottutu+nnaanu

Kottu=verb nnanu=Tense+PNG

1. Habitual present and future tense:

In this the word ending with the roots : taanu,taavu,taaru,taadi,taadu,taamu.

Ex:kottu+taanu.

1. Past tense:

In this the word ending with the roots :

tini,tivi,tiri,timi,tiri,…..etc.

Ex:kottu+tini.

1. **ADVERB - RB:**

As we will see, adverbs often tell when, where, why, or under what conditions something happens or happened. Adverbs frequently end in *-gA*; however, many words and phrases not ending in *-ly* serve an adverbial function and an *–gA and –ayina* ending is not a guarantee that a word is an adverb. The words *andamgA*, nallagA for instance, are adjectives:

The representation of adverb is ADJ meVllagA, woVMxaragA

1. **Negation - NEG**:

By watching suffixes Kaadu, kuDadu, ledu, kudarDu we can say the tagging the word NEG.

1. **Conjunction - CC**:

The conjunctions in Telugu language are mariyu(and), leda(or). For this words the tagging word representation is CC.

Conjunction:

* CONJ mariyu
* CONJ leka
* CONJ ani
* CONJ kanI
* CONJ gAnI

1. **CODING:** The coding of the program is:

#!/usr/bin/env perl

use strict;

use warnings;

use utf8; #used for encoding and decoding purpose

#declaring var i and arrays

my $i = 0;

my @words = ();

my @vocab\_list = ();

my $infile = shift or die "please provide an input file name!\n";

open my $IN,"<encoding(utf8)",$infile or die "unable to open the input file $infile!\n";

my $outfile = shift or die "please provide an output file name!\n";

open my $OUT,">encoding(utf8)",$outfile or die "unable to open the output file $outfile!\n";

while (my $line = <$IN>) {

chomp $line;

next if ($line =~ /^\s\*$/);#tokonization

for($line){

s/<\/?.+?>//g;

s/\'(.+?)\'/$1/g;

s/[!.,::?()"\/]//g;

s/\s~\s//g;

s/[a-z\/]//g;#deleting small alphabets

s/[A-Z\/]//g;#deleting caps

s/[0-9\/]//g;#deleting numeric data

s/^\s\*//;

s/\s+$//;

s/\'//;

}

my @element = split/\s+/,$line;#spliting

push @words,@element;

}

foreach my $word (@words) {

#assign input data to each operation in loop and checks true or false

if($word =~ /(\w+)కం$|(\w+)గ్నం$|(\w+)రం$|(\w+)గం$|(\w+)హం$|(\w+)ర్వం$|(\w+)వ్యం$|(\w+)ర్మ$|(\w+)దం$|(\w+)స్యం$|(\w+)ల్య$|(\w+)చెం$|(\w+)ర్యం$|(\w+)టం$|(\w+)ల$|(\w+)ష్టం$/&$word !~/(\ w+)న్ని$|(\w+)మ్మ్ర$/){#pattern matching

print $OUT "$word NN\n";

}

if($word =~ /(\w+)లొ$|(\w+)కి$|(\w+)లు$|(\w+)కు$|(\w+)తొ$/){ #pattern matching

print $OUT "$word NN\n";#print output

}if ($word =~ /(\w+)కి$/){

print $OUT "$word IN\n";

}if($word =~ /(\w+)తొ$/){

print $OUT "$word IN\n";

}if($word =~ /(\w+)కు$/){

print $OUT "$word IN\n";

}if($word =~ /(\w+)లొ$/){

print $OUT "$word IN\n";

}if($word =~ /(\w+)యొక్క$/){

print $OUT "$word IN\n";

}if($word =~ /(\w+)లోకి$/){

print $OUT "$word IN\n";

}if($word =~ /(\w+)వైపుకు$/){

print $OUT "$word IN\n";

}

if ($word =~ /(\w+)చాడు$|(\w+)చారు$|(\w+)చావు$|(\w+)చాను$/){ #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)టాడు$|(\w+)టాను$|(\w+)టాడు$|(\w+)టాము$|(\w+)టారు$|(\w+)టవి$/) { #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)చ్చారు$|(\w+)చ్చాడు$|(\w+)చ్చును$/) { #pattern matching

print $OUT "$word VB+Tense+PNG\n\n";

}

if ($word =~ /(\W+)ట$|(\W+)చు$|(\W+)యు$|(\w+)డం$|(\w+)టం$/&$word !~/(\w+)డ్డం$|(\w+)ష్టం$|ట్టం$/&$word !~/\bఖం(\w+)\b/){ #pattern matching

print $OUT "$word VB\n";

}

if ($word =~ /(\w+)నది$|(\w+)నావు$|(\w+)నాను$|(\w+)నవి$|(\w+)నారు$|(\w+)నాడు$/) { #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)స్తుంది$/){ #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)తుంది$|(\w+)టుంది$/){ #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)యింది$/){ #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)న్నది$|(\w+)న్నవి$/) { #pattern matching

print $OUT "$word VB+Tense+PNG";

}

if ($word =~ /(\w+)న్నాను$|(\w+)న్నావు$|(\w+)న్నది$|(\w+)న్నాము$|(\w+)న్నారు$|(\w+)న్నవి$|(\w+)న్నాడు$/) { #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)తాను$|(\w+)తావు$|(\w+)తాము$|(\w+)తారు$|(\w+)తావి$|(\w+)తాడు$/) { #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)దును$|(\w+)దువు$|(\w+)దుము$|(\w+)దురు$|(\w+)తిమి$|(\w+)తిరు$|(\w+)తివి$|(\w+)తిని$/) { #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)స్తాను$|(\w+)స్తావు$|(\w+)స్తాము$|(\w+)స్తారు$|(\w+)స్తావి$|(\w+)స్తాడు$/) { #pattern matching

print $OUT "$word VB+Tense+PNG\n";

}

if ($word =~ /(\w+)స్తుంది$/){ #pattern matching

print $OUT "$word VB+Tense+PNG";

}

if ($word =~ /(\w+)గా$|(\w+)కు$/){

print $OUT "$word JJ\n";

}

if ($word =~ /(\w+)వాడు$|(\w+)మైన$|(\w+)వైన$/){

print $OUT "$word JJ\n";

}

if($word =~ /(\w+)పైన$/){

print $OUT "$word IN\n";

}if($word =~ /\bలేదా\b/){

print $OUT "$word CC\n";

}if($word =~ /\bమరియు\b/){

print $OUT "$word CC\n";

}if($word =~ /\bకాదు\b/){

print $OUT "$word NEG\n";

}if($word =~ /\bలేదు\b/){

print $OUT "$word NEG\n";

}

if($word =~ /!/){

print $OUT "$word EXLAMETARY\n";

}

if ($word =~ /(\w+)ప్పుడు$|(\w+)క్కడ$|(\w+)లేదు$|(\w+)గా$/) { #pattern matching

print $OUT "$word RB\n";

}

if($word =~ /\bబాగా\b/){

print $OUT "$word RB\n";

}

if($word =~ /\bబయట\b/){

print $OUT "$word RB\n";

} if($word =~ /\bచాలా\b/){

print $OUT "$word RB\n";

}

}

close $IN;

**Programming design:**

**Discussion:**

In this section I can do only two steps:

They are

1. Splitting and tokenization.

2. Automatic extraction of Nouns, verbs, Adjectives, Adverbs, preposition, Pronoun, exclamatory based on roots of words and Conjunction, Negations are directly taken from programme.

***Explanation***:

Splitting can be done on following logic:

@element = split/\s+/,$line;

Tokenization can be done on following logic:

next if ($line =~ /^\s\*$/);

for($line){

s/[A-Z\/]//g;

s/[0-9\/]//g;

s/\s~\s//g;

s/^\s\*//;

s/\s+$//;

s/\'//;

s/\-//;

s/\=//;

}

After tokenization, By using regular expressions program splits the sentences in to words, to do this operation if any space is there between words it will divided into separate words.

**Matching Operation using loops:**

For matching operation while and if else loop and regular expressions is required.

While is used for assigning the input line to each operation in loop.

If loop will first check the condition if the condition is true that will continue the checking process, otherwise it terminate the loop and it will go to the else condition.

***Use utf8:***

This function will be encode the data while executing the program and decode the same data while printing the output. Without utf8 non ASCII character will be encoded.

**The logic of Regular expression:**

***$line =~ /(\w+)చాడు$|(\w+)చారు$|(\w+)చావు$|(\w+)చాను$/***

#matching and extract the root words chaadu,chaaru etc.

$line =~ /(\w+)టాడు$|(\w+)టాను$|(\w+)టాడు$|(\w+)టాము$|(\w+)టారు$|(\w+)టవి$/

#matching and extract the root words taamu,taadu etc.

$line =~ /(\w+)చ్చారు$|(\w+)చ్చాడు$|(\w+)చ్చును$/) #pattern matching

my $root2 = substr$line,0,-7; #extracting

$base3 = $root2."వ్వటం"; #adding root

Remaining part of the program can explained on same logic(shown on above).

1. **Sample output of POS tagger:**

కోపంగా RB

అంగీకరిస్తున్నట్లు NN

ఇష్టం NN

ఊపడం VB

మాటలు NN

వేగంగా JJ

వేగంగా RB

వెటకారంగా JJ

వెటకారంగా RB

పలకడం VB

పిచ్చిగెంతులు NN

పిల్లికూతలు NN

హాస్యానికి NN

హాస్యానికి IN

హాస్యానికి NN

హాస్యానికి IN

మొదటివాడు JJ

మరణానంతరం NN

విధానాల NN

ఉనికిలోకి NN

ఉనికిలోకి IN

ప్రపంచబ్యాంకు NN

ప్రపంచబ్యాంకు IN

ప్రపంచబ్యాంకు JJ

విధానాలు NN

ఆర్దికంగా JJ

ఆర్దికంగా RB

స్వాతంత్య్రానికి NN

స్వాతంత్య్రానికి IN

ఉద్యమాల NN

పాలకుల NN

మెడలు NN

అమల్లోకి NN

అమల్లోకి IN

చర్యలు NN

ఒక్కటొక్కటిగా JJ

ఒక్కటొక్కటిగా RB

ప్రచారంలోకి NN

ప్రచారంలోకి IN

ప్రచారంలోకి IN

తెచ్చారు VB+Tense+PNG

సిద్దాంతాలుగా JJ

సిద్దాంతాలుగా RB

కమ్యూనిస్టుల NN

వర్గాలకు NN

వర్గాలకు IN

వర్గాలకు JJ

బాగా JJ

బాగా RB

బాగా JJ

పాలకపార్టీలుగా JJ

పాలకపార్టీలుగా RB

పార్టీలు NN

సంకీర్ణప్రభుత్వాల NN

యుగం NN

పాలకవర్గాల NN

ఏమాత్రం NN

సడల్లేదు RB

పైగా JJ

పైగా RB

ముందుకు NN

ముందుకు IN

ముందుకు JJ

ప్రజల NN

సమస్యల NN

ఉద్యమాలపైకి IN

ముందుకు NN

ముందుకు IN

ముందుకు JJ

విస్తరించుకున్నారు VB+Tense+PNG

వారికి IN

కల్పించారు VB+Tense+PNG

1. **HYPOTHESIS:**

* In Telugu i found many Hypotheses in Output of POS Tagger. Telugu is morphologically complex we face the problems with the case markers.
* Especially some case markers will come with Adjectives, Adverbs.
* Some case markers i found in my project common for Preposition, Nouns, at that conditions programming is giving double tagging like single word tagged under two POS categories, i.e NN and PP, ADV,ADJ.

-ki suffix is occured in Nouns and Prepositions so it is giving two POS tagging sets.

Ex: rajesh ki fever vatchinDi. In this sentence ki is occurred in Nouns category.

Rajesh got fever.

. Rajesh IkkaDiki vachaDu. Rajesh came here.

In the above sentence ki is in preposition position.

So the POS tagger giving the output as NN and PP.

Common suffixes are found in various categories of parts of speech tagging .

In the similar manner for Adjectives and Adverbs also.

1. **Limitations:**

* This tagger will tag the words depends upon affixes and words, but it won’t check the fallowing and preceding words in the sentence, so that it won’t give proper output.
* One more drawback is there suppose ki, to, ku etc... suffixes are there in a word it is not only taking that suffixes, it is also taking vowel soun of that words, so it is confusing to tag, and giving multiple tagging in the output.

1. **CONCLUSION:**

This POS tagger will give the output 61% of accuracy, which is much better then others. It is applicable to extra large text for more then one million words and Depending upon system configuration it will load the large files. Before loading text file make sure that clarify without grammatical mistakes, if it is there Tagger will give wrong POS tagging. By this programming code we can develop web application in further.

1. **Bibliography:**

* <http://www.cs.cf.ac.uk/Dave/PERL/>
* <http://www.tutorialspoint.com/perl/perl_regular_expressions.htm>
* BH. Krishnamurthi and J.P.L Gwynn, "A Grammar of Modern Telugu", Oxford University Press, New Delhi, 1985.
* G Bharadwaja Kumar, Kavi Narayana Murthy, and B B Chaudhuri, "Statistical Analysis of Telugu Text Corpora", In International Journal of D.
* Kavi Narayana Murthy and Badugu Srinivasu, "On the Design of a TagSet for Dravidian Languages", In 40th All India Conference of Dravidian Linguists, University of Hyderabad, Hyderabad, India, 18-20 JUNE-2012.
* <http://www.academia.edu/3020912/A_TELUGU_MORPHOLOGICAL_ANALYZER>
* Assessment and development of POS tag set for Telugu. Asian language paper.
* Programming for linguistics: perl for language researchers by Michael Hammond.