Multi-Language
Text Pre-processor User Guide

Jim Cowie and Ahmed Abdelali

MCCS-04-332

Computing Research Laboraroy
Box 30001
New Mexico State University
Las Cruces, NM 88001

The Computing Research Laboratory was established by the
New Mexico State Legislature
under the Science and Technology Commercialization Commission
as part of the Rio Grande Research Corridor

Contents

Multi-Language Text Pre-processor User Guide

Version 4.0: May 10th, 2004

Abstract

This paper describes the operations carried out by the Multi-language preprocessor. This takes raw text in Arabic, English, and Persian after it has been cleaned of markup (by an HTML, SGML, or other appropriate parser) and carries out tokenization, morphological analysis, lexical lookup and pattern recognition of basic text elements. If HTML is present then it is treated as white-space. The system uses a part of speech lexicon, onomastica (proper name dictionaries) for places, organization, and person names. These lists include complete names and also text element components, which can be used to recognize novel names. Number phrases and dates are also handled using patterns of text component types.

The output of the preprocessor is a set of span descriptors (multiple are possible for each text token or set of tokens) and an associated set of properties. Ambiguity is preserved to a degree at this point and will be resolved in the subsequent syntactic and semantic analysis. The span descriptors can be output as text or stored as annotations on a document held in a document collection.

The major novelty of the approach adopted here is that lexical lookup occurs only once for each distinct element in the text (i.e. each word form is only looked up once). This is intended partly for speed of processing and efficient use of storage. The principle advantage, however, is that the *document lexicon* produce by this lookup process contains the frequency of each token and of each morphological variant. Thus the pre-processor produces as a by-product of its operation the indices, counts, and locations in the document of tokens and sets of tokens, which are ready to be used by document indexing and summarization systems.

Introduction

In the example below the output of analysis consists of a span, the root form used for lookup, the original text token and a set of properties. The list of properties is given in table form in Appendix A.

Example

Input Sentence:

The communist leaders of remote Laos are loosening their grip on the country in a local version of the Kremlin's sweeping reforms.

Output Results:

((3,5), root='the',post=art,case=capitalized)
((3,5), root='the',post=adv,case=capitalized)
((7,15), root='communist',post=adj,case=lower)

```
((7,15), root='communist',post=n,case=lower)
((17,23), root='leader',post=n,number=plural,case=lower)
((25,26), root='of',post=prep,case=lower)
((28,33), root='remote',post=adj,case=lower)
((35,38), root='laos',post=pn,type=country,case=capitalized)
((40,42), root='are',case=lower)
((44,52), root='loosen',post=v,form=participle,case=lower)
((54,58), root='their',post=art,case=lower)
((60,63), root='grip',post=n,case=lower)
((65,66), root='on',post=adj,case=lower)
((65,66), root='on',post=adv,case=lower)
((65,66), root='on',post=prep,case=lower)
((68,70), root='the',post=art,case=lower)
((68,70), root='the',post=adv,case=lower)
((72,78), root='country',post=adj,case=lower)
((72,78), root='country',post=n,case=lower)
((80,81), root='in',post=adj,case=lower)
((80,81), root='in',post=adv,case=lower)
((80,81), root='in',post=n,case=lower)
((83,83), root='a',post=art,case=lower)
((83,83), root='a',case=lower)
((85,89), root='local',post=adj,case=lower)
((85,89), root='local',post=n,case=lower)
((91,97), root='version',post=n,case=lower)
((99,100), root='of',post=prep,case=lower)
((102,104), root='the',post=art,case=lower)
((102,104), root='the',post=adv,case=lower)
((106,112), root='kremlin',post=pn,type=city,case=capitalized)
((106,112), root='kremlin',post=pn,type=company,case=capitalized)
((113,113), root='\'',post=punct)
((114,114), root='s',case=lower)
((116,123), root='sweeping',post=adj,case=lower)
((125,131), root='reform',post=n,number=plural,case=lower)
((125,131), root='reform',post=v,tense=present,person=third,case=lower)
((132,132), root='.',post=punct)
```

Stages of Processing

The text is initially analyzed by a tokenization step, which recognizes basic token types, inserts a representation in a hash table and produces a span descriptor for the token. Thereafter all processing is carried out either on the hash table, for morphology and lexical lookup, or on the span descriptors for phrasal and pattern recognition.

Installing and Running

A- From Command Line

The text preprocessor is supplied as a zip file. This normally unpacks to the directory C:\Text-preprocessor-v4.0a. If another directory (<PREP-DIR>) is chosen the file <PREP-DIR>\Projects\prep\dicts.h will require editing to show the current location of the system lexicons and the project workspace <PREP-DIR>\Projects\prep\prep.dsw loaded into Microsoft Visual C++ and the program prep rebuilt. A more verbose output can be obtained by defining VERBOSE as 1 before compiling the program.

The preprocessor program “prep” can be run in any directory. A directory <PREP-DIR>

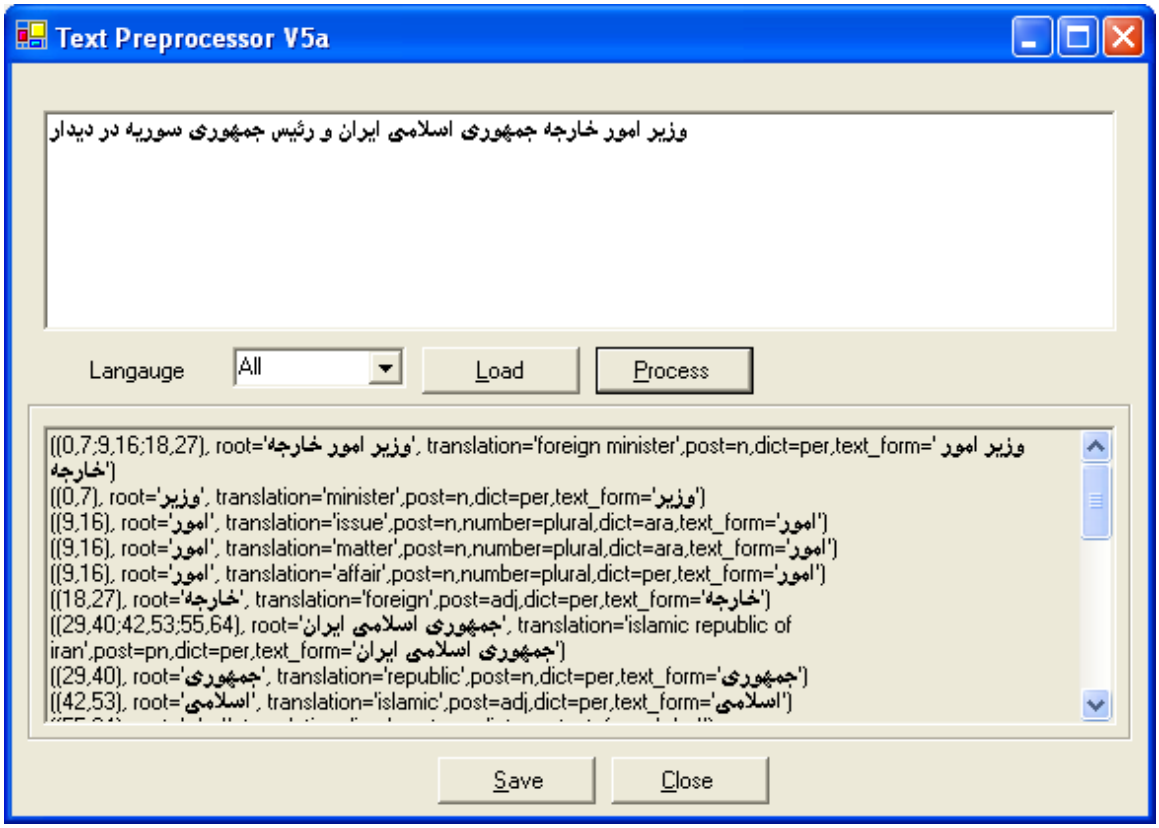
contains sample text files and command files to run the program, which reads input from stdin and writes to stdout. All error and information messages are written to pre-logfile.txt in the directory from which prep is run.

e.g.

prep <one-sentence.txt >one-sentence_out.txt
prep [arabic|persian] <one-sentence.txt >one-sentence_out.txt

B- From GUI

The GUI allows the user to open any utf-8 encoded file or the user could type the text he want to process. The languages could be processed by the current version are Arabic, English and Persian. The Language Combo box allows the user to specify the language, which will be used for processing the text. –All option will analyze the text with all the three different analyzers.
The user fires the analysis task by clicking on "Process". The result will be displayed on the lower text area.
The user can save the result analysis using the button "Save".



Modifying the Lexicons

The system currently used six lexicon files; one contains parts of speech, three contain places, personal name units, and company name units, one contains irregular forms, and the last contains mostly oddball stuff which requires additional information for processing (e.g. month names, initial letters). All the files are composed of a search string in lower case terminated with a colon followed by one or more information units which are comma separated. The information units can be a part of speech, or a complex set of type, value pairs enclosed in brackets and separated internally by semi-colons. These are described in detail below (Lexicon Format).

The search strings are space separated lower case tokens as produced by the system tokenizer. Thus all punctuation in a string must be delimited by spaces

e.g.

baby carriage:
e . g .:
hold – up:

All characters apart from alphabetic and numeric are treated as individual tokens. Mixed alphanumeric strings are also recognized as single tokens.

e.g.

b52 bomber:n
23rd cavalry:n

There are currently six lexicon files. In general the order in which the files are looked up is immaterial. However, if the irregular lexicon file contains roots, which do not appear elsewhere in a text, then it should be looked-up first in the list as the root will then appear in the word list, which allows phrasal items to be recognized.
The six files are –

- irregular-lex: contains common irregular forms
- pos-lex: contains regular parts-of-speech
- all-companies: complete company names, company name components
- all-names: first and family names, titles, and name suffixes
- all-places: complete place names, name start words (e.g. North), name termination words (e.g. Lake, City)
- all-other-terms: date related words (month etc.), initials

An index file accompanies each file. This indicates the start point (block) for a reduced set of character sequences. A program called “lookup” builds this index automatically. The programs to build the index file are found in the folder “work-directory”. A command file, “index_lex,bat”, which ensures file, “lexicon”, is in lower case, sorts it, and builds an index file, “lexicon-sort”, is located in the same directory. For each lexicon file there needs to be an accompanying “-sort” file (e.g. all-names and all-names-sort). The lexicon files can be edited, taking care to maintain alphabetic order, without any need to create a new index file. If lookups fail to find a new term, however, then it is

probably time to copy the file to “lexicon”, run “index_lex” and then copy “lexicon” and “lexicon-sort” back to the corresponding lexicon file names. You should keep backup copies of all the lexicon files at regular intervals, just in case an update process fails, or a file is otherwise damaged.

Lexicon Format

There are two forms for a lexical entry. The most common contains the lexical string, a colon, and one or more parts-of-speech separated by commas.

A word with multiple parts of speech can be written on one line or on multiple lines.
Thus –

dog:n, v

and

dog:n
dog:v

are equivalent. Don’t forget all tokens apart from alphanumeric strings must be separated by spaces e.g. “e . g .”

The second form, which can be mixed with the first, allows all the possible property values to be over-ridden. It also allows the specification of a root form (lemma) for a word. This allows the root to be recognized in compound terms in which it occurs in a modified form. The root is also used stored as the root to be used in further processes.
Thus –

was:(“be”;post=verb;tense=past)
i . b . m .:(“international business machines”;post=pn;proper=company)

A common error is to use a comma separator in these property lists. An extended BNF definition of an entry is given below along with tables of currently recognized values. Examples of inputs can be found in the lexicon files.

BNF

lexical_entry: word(“ “word)*”:”feature_list(“,”feature_list)*
word:([a-z0-9]+)|([^a-z0-9:])
feature_list: group!pos
pos: <any of the parts of speech listed below>
group:”(“value(“;”value)*”)”
value:root|pair
root:””’[^”]+””’
pair:group_name”=”group_value
group_name: <any of the groups listed below, some make more sense than others!>

group_value:<any of the group values listed below, a group value should be selected from the group specified in the group_name before the "=" sign>

Group Names and Associated Values

Most of these groups are only of concern to the inner workings of the pre-processor. To define irregular forms of words the most important groups are – POST, TENSE, NUMBER, PERSON and DEGREE.

Group	**Possible Values**
POST	Part of Speech (derived from lexicon + morphology + pattern recognition)
	ADJ, ADV, CONJ, DET, INTERJ, NOUN, PREP, PRON, VERB, NUM, PUNCT, PN, AUX, OTHER, TO, DATE, TIME Following short forms also accepted: N, V
PROPER	Proper name components (derived from lexicon +pattern recognition)
	BEGCOMP – Company name beginning – e.g. Importadora, Eastern BEGPLACE – Place name beginning – e.g. Lake, North CITY – City name – e.g. Rio de Janeiro COMFIX – Company suffix – e.g. Ltd., GmBH COMPANY – Complete company name – e.g. Toyota Motor Company Ltd. COMPRE – Company prefix – e.g. PT, Department of COUNTRY – Country name – e.g. China ECW – Ending company word – e.g. Airlines, Laboratories, University EPW – Ending place word – e.g. island, coast, bay FAMNAME – Surnames FORNAME - Firstnames PROVINCE – Provinces and states TITLE – Personal titles – e.g. major, sir, dr. NAME – Complete person name FAMFIX – Name suffix – e.g. II, Jnr.
TENSE	Tense (derived from lexicon +morphology)
	PRESENT, PROGRESSIVE, IMPERFECT, PAST, FUTURE, PRESENT_PARTICIPLE, PAST_PARTICIPLE
NUMBER	Number (derived from lexicon +morphology)
	SINGULAR, PLURAL
PERSON	Person (derived from lexicon + morphology)
	FIRST, SECOND, THIRD

DEGREE	Degree (derived from lexicon + morphology)
	POSITIVE, COMPARATIVE, SUPERLATIVE
OTHER_FEATURES	Properties used by patterns (derived from lexicon)
	MONTH, TIME_OBJECT, INITIAL_LETTER
CASE	Character case (derived from tokenization and used by pattern recognition)
	UPPER, LOWER, NUMERIC, DECIMAL, ALPHANUMERIC, NUMERICALPHA, ORDINAL, BLANKLINE, STOP, HYPHEN, SLASH, BACKSLASH, COLON, SEMICOLON, APOSTROPHE , QUESTIONMARK, EXCLAMATIONMARK, DIERISES, CAPITALIZED, MIXED, COMMA, OPEN_ROUND, CLOSE_ROUND, OPEN_WAVY, CLOSE_WAVY, OPEN_ANGLE, CLOSE_ANGLE, OPEN_SQUARE, CLOSE_SQUARE
LOOKUP	Classes of tokens (derived from tokenization)
	WORD, NUM, ANUM, PUNCT, SPACE, SYMBOL, BRACKET, SEPARATOR, QUOTE
MORPH	Suffix changes (derived from morphology)
	NO_FIX, S_FIX, ED_FIX, ING_FIX, ER_FIX, LY_FIX, EST_FIX, I_FIX, ISH_FIX, NESS_FIX, IRREGULAR
LEXPOST	Parts of speech (derived from lexicon only)
	ADJ, ADV, CONJ, DET, INTERJ, N, PREP, PRON, V

Arabic Morphology

The Arabic Morphology module uses a simple approach of dividing the Arabic word into three parts
Prefix: consist of as many as four concatenated prefixes, or could be null
Stem: it is composed of root and pattern morphemes
Suffix: consist of as many as three concatenated suffixes, or could be null

Because the Arabic prefixes and suffixes are finite number, their respective lexicons could be considered complete, the Stem lexicon needs
Using the approach the word "wsyktbwnhA" (See **Appendix B** for Arabic Transliteration) would be analyzed as follows:

Prefix	**Stem**	**Suffix**
wsy	ktb	wnhA

The tree-part approach entails the use of three lexicons: Prefixes lexicon, Stem lexicon, and Suffixes lexicon. For a word to be analyzed its parts must have an entry in each lexicons. Assuming the both null prefix and null suffix are both possible. Here some example of valid words:

Prefix	Stem	Suffix
Al	ktAb	(null)
(null)	ktAb	An
wAl	ktAb	yn
y	ktb	(null)
t	ktb	yn
st	ktb	hA

Any combinations of Prefix-Stem-Suffix is not necessarily valid or a legal word. In order to confirm if the composition Prefix-Stem-Suffix is a valid Arabic word Morphological categories were assigned to each entry in the lexicons (See PREFIX/STEM/SUFFIX CATEGORIES Tables below).
The category assignment was based on compatibility and the part of speech of the Suff/Stem/Prefix
When a prefix, stem, suffix found in their respective lexicons, the morphology analysis extract the morphological category assigned to every one and than use the categories for a validation process. The validation process uses three truth tables Prefix-Stem, Stem-Suffix, and Prefix-Suffix each tables is a two dimensional array of 0 or 1, 0 indicate incompatibility and 1 compatibility once the combination checked and confirmed valid the morphological analyzer output the valid combination (Prefix-Stem-Suffix) that composes the word.

Example

Input Sentence:

و كانت الوفود المشاركة من البلدان الإفريقية والعربية

Output Results:

((0,1), root='و', translation='and',post=other,text_form='و')
((3,10), root='كان', translation='was',post=v,tense=past,text_form='كانت')
((12,23), root='وفود', translation='delegation ',post=n,number=plural,pref=det,det=the,text_form='الوفود')
((25,40), root='مشارك', translation='participant',post=n,subtype=female,pref=det,det=the,text_form='المشاركة')
((25,40), root='مشارك', translation='association',post=n,subtype=female,pref=det,det=the,text_form='المشاركة')
((25,40), root='مشارك', translation='participation',post=n,subtype=female,pref=det,det=the,text_form='المشاركة')
((42,45), root='من', translation='from',post=other,text_form='من')
((42,45), root='من', translation='afflict',post=v,text_form='من')
((47,60), root='بلدان', translation='country ',post=n,number=plural,pref=det,det=the,text_form='البلدان')
((47,60), root='بلد', translation='country',post=n,number=dual,subtype=male,pref=det,det=the,text_form='البلدان')
((62,79), root='إفريقي', translation='african',post=n,subtype=female,pref=det,det=the,text_form='الإفريقية')
((81,96), root='عربي', translation='arab',post=n,subtype=female,pref=conj_det,conj=and,det=the,text_form='والعربية')

These are the descriptive names for categories

Prefix value	Description	Details
conj	Conjunction	and
conj	Conjunction	so
conj_det	Conjunction + Determinant	and the
conj_det	Conjunction + Determinant	so the
conj_para	Conjunction + Para-phrase	and in order to
conj_prep	Conjunction + Preposition	and like
conj_prep	Conjunction + Preposition	so in/with
conj_prep	Conjunction + Preposition	so like
conj_prep_det	Conjunction + Preposition + Determinant	and in/with the
conj_prep_det	Conjunction + Preposition + Determinant	so in/with the
conj_prep_det	Conjunction + Preposition + Determinant	so like the
det	Determinant	the
para	Para-phrase	in order to
prep	Preposition	in/with
prep_det	Preposition + Determinant	in/with the
prep_det	Preposition + Determinant	like the

Arabic Lexicon

The Arabic Lexicons are arranged the same way for the English lexicons with the different part of speech and morphological categories, the only exception is the two additional features Arabic Morphological categories and Translation.
The Arabic morphological categories were defined based on word part of speech and the combinations of Prefixes/Suffixes. Details about the categories are in **Appendix B**.
The English translation for the word is presented between the both <…>

Example of Arabic lexicon:

مـــذابيع:(aracode=r035;number=plural;<transmitter>)
مذب:(aracode=r017;<fly swatter>)
مـذبح:(aracode=r009;<slaughterhouse>)
مـذبح:(aracode=r017;<massacre>)
مـذبح:(aracode=r017;<slaughter>)
مذبـذب:(aracode=r001;<fluctuating>)
مذبـذب:(aracode=r001;<wavering>)
مذبـذب:(aracode=r005;<oscillator>)

Persian Morphology

We followed the same approach for Arabic; the Persian word was decomposed to the three components Prefix, Suffix, and Stem. Using the combination we assigned the possible part of speech to the valid stem.

The list of combination Prefix-Stem-Suffix are listed in the **Appendix C**

Example

Input Sentence:

وزیر دفاع آمریکا روز یکشنبه به افغانستان سفر میکند.

Output Results:

((0,7;9,16), root='وزیر دفاع', translation='minister of defense',post=n,text_form='وزیر دفاع')
((0,7;9,16), root='وزیر دفاع', translation='secretary of defense',post=n,text_form='وزیر دفاع')
((0,7), root='وزیر', translation='minister',post=n,text_form='وزیر')
((9,16), root='دفاع', translation='defense',post=n,text_form='دفاع')
((9,16), root='دفاع', translation='defense',post=n,text_form='دفاع')
((18,29), root='آمریکا', translation='america',post=pn,text_form='آمریکا')
((31,36;38,49), root='روز یکشنبه', translation='sunday',post=num,text_form='روز یکشنبه')
((31,36), root='روز', translation='day',post=n,text_form='روز')
((31,36), root='روز', translation='daytime',post=n,text_form='روز')
((31,36), root='روز', translation='day',post=n,text_form='روز')
((31,36), root='روز', translation='daytime',post=n,text_form='روز')
((38,49), root='یکشنبه', translation='sunday',post=n,text_form='یکشنبه')
((38,49), root='یکشنبه', translation='sunday',post=num,text_form='یکشنبه')
((51,54), root='به', translation='for',post=prep,text_form='به')
((51,54), root='به', translation='for',post=prep,text_form='به')
((56,73), root='افغانستان', translation='afghanistan',post=pn,text_form='افغانستان')
((75,80;82,91), root='سفر کردن', translation='travel',post=v,text_form='سفر کردن')
((75,80;82,91), root='سفر کردن', translation='travel',post=v,tense=past,text_form='سفر کردن')
((75,80), root='سفر', translation='travel',post=n,text_form='سفر')
((75,80), root='سفر', translation='trip',post=n,text_form='سفر')
((75,80), root='سفر', translation='travel',post=n,text_form='سفر')
((75,80), root='سفر', translation='trip',post=n,text_form='سفر')
((82,91), root='کردن', translation='do',post=v,text_form='میکند')
((92,92), root='.',post=punct,text_form='.')

Persian Lexicon

In the same manner the Persian Lexicon is defined the same way as the Arabic, Example of Persian lexicon:

یمـرتض:(post=pn;<morteza>)
مـرتع:(post=noun;<pasture>)
مـرتفع شـدن:(post=verb;<elevate>)
مـرتفع شـدن:(post=verb;voice=passive;<eliminate>)
مـرتفع کـردن:(post=other;<elevate>)
مـرتفع کننـده:(post=noun;<lifter>)
مـرتفع:(post=adj;<high>)
یاگـروگ‌ی مـرتهن:(post=noun;<avuncular>)
مـرتهن:(post=adj;<mortgagee>)

Appendix A. Text Pre-processor Part of Speech Features

Table 1. Features supplied by Tokenization, Morphology, and Lexical Lookup.

Token Type	Token Class	Morphology	Part of Speech	PN Properties
Word	Upper	Unchanged	Adjective	
	Lower	S ending	Adverb	
	Capitalized	ED ending	Conjunction	
Number	Numeric	ING ending	Determiner	
	Decimal	ER ending	Interjection	
Alpha Numeric	Alpha Number	LY ending	Noun	
	Number Alpha	EST ending	Preposition	
	Ordinal	ISH ending	Pronoun	
Space	Whitespace	I ending	Verb	
	Blank line	NESS ending	Other	
Punctuation	Stop		Proper Noun	Begin Company
	Hyphen			Begin Place
	Slash			City
	Backslash			Company Suffix
	Colon			Company Name
	Semi-colon			Country
	Double Quote			End Company
	Apostrophe			End Place
	Question Mark			Family Name
	Exclamation			Fore Name
	Dierises			Person Name (label is name)
	Comma			Province
Symbol	Other			Title

Table 2. Features supplied by combining Morphology and Part of Speech Features.

Tense	Person	Number	Polarity
Present	First	Singular (default)	Positive (default)
Progressive	Second	Plural	Negative
Imperfect	Third		
Past			
Future			
Present Participle			
Past Participle			

Appendix B. Arabic Part of Speech Features and Categories

Arabic Transliteration Table

ء '	ذ *	ل l
آ |	ر r	م m
أ >	ز z	ن n
ؤ &	س s	ه h
إ <	ش $	و w
ئ }	ص S	ى Y
ا A	ض D	ي y
ب b	ط T	ً F
ة p	ظ Z	ٌ N
ت t	ع E	ٍ K
ث v	غ g	َ a
ج j	ـ _	ُ u
ح H	ف f	ِ i
خ x	ق q	ّ ~
د d	ك k	ْ o

Arabic Morphological Categories

PREFIX CATEGORIES

001 nulp;	010 imvp; imperfect verb prefix
002 conj; conjunction	011 imvp; imperfect verb prefix
003 prep; preposition	012 imvp; imperfect verb prefix
004 prep; preposition	013 imvp; imperfect verb prefix
005 defi; definite article	014 imvp; imperfect verb prefix
006 ppda; preposition + definite article	015 imvp; imperfect verb prefix
007 ppda; preposition/particle + definite article	016 imvp; imperfect verb prefix
008 imvp; imperfect verb prefix	017 imvp; imperfect verb prefix
009 imvp; imperfect verb prefix	

STEM CATEGORIES

001..025	Common noun stems without orthographic change
026..055	Common noun stems with orthographic change
056..057	Function words
058..059	Proper noun stems
060..081	Perfect verbs stems
082..088	Perfect/Imperfect verb stems
089..129	Imperfect verb stems

SUFFIX CATEGORIES

001 nuls; null suffix
002 fmdl; feminine dual
003 fmdl; feminine dual
004 fmdl; feminine dual
005 fmdl; feminine dual
006 fmsl; feminine singular
007 fmsl; feminine singular
008 msdl; masculine dual
009 msdl; masculine dual
010 msac; masculine accusative
011 msdl; masculine dual
012 msdl; masculine dual
013 fmpl; feminine plural
014 fmpl; feminine plural
015 pspn; possessive pronoun
016 mspl; masculine plural

017 mspl; masculine plural
018 pspn; possessive pronoun
019 msdl; masculine dual
020 mspl; masculine plural
021 mspl; masculine plural
022 dopn; direct object pronoun
023 pvsf; perfect verb suffix
024 dopn; direct object pronoun
025 pvsf; perfect verb suffix
026 dopn; direct object pronoun
027 pvsf; perfect verb suffix
028 dopn; direct object pronoun
029 msdl; masculine dual
030 mddo; masculine dual + direct object pronoun
031 mspl; masculine plural
032 mpdo; masculine plural + direct object pronoun
033 dopn; direct object pronoun
034 dopn; direct object pronoun
035 fmpl; feminine plural
036 fpdo; feminine plural + direct object pronoun
037 fpdo; feminine plural + direct object pronoun
038 dual; dual
039 ddop; dual + direct object pronoun
040 dual; dual
041 ddop; dual + direct object pronoun
042 mspl; masculine plural
043 mpdo; masculine plural + direct object pronoun
044 mpdo; masculine plural + direct object pronoun
045 mspl; masculine plural
046 mpdo; masculine plural + direct object pronoun
047 mpdo; masculine plural + direct object pronoun
048 fmsl; feminine singular
049 fsdo; feminine singular + direct object pronoun
050 fmsl; feminine singular
051 fsdo; feminine singular + direct object pronoun

Appendix C. Persian Part of Speech Features and Categories

Persian Verb Topics

- Citation form of the Persian verbs in the dictionary are basically infinitival form of the verbs.

- **Present Stem** is formed by removing the "b" from the imparative form of the verb. Present stems are specified in the dictionary for the verbs.
  ex.:
  Citation form: frvxtn;
  PresentStem = bfrv^s – b = frv^s;

- **Past stem** can be derived from removing "n"from the end of infinitive.
  ex.:
  Citation form = frvxtn;
  PastStem = frvxtn – n = frvxt

1. Personal Inflections for Verbs

1.1 Present inflection (verb:rftn, present stem:rv*) Prefix 'my' &* Suffixes

m my + rv + m = myrvm (person:first; number:singular)
y my + rv + y = myrvy (person:second; number:singular)
d my + rv + d = myrvd (person:third; number:singular)
ym my + rv + ym = myrvym (person:first; number:plural)
yd my + rv + yd = myrvyd (person:second; number:plural)
nd my + rv + nd = myrvnd (person:third; number:Plural)

1.2 Past inflection (verb:rftn [go]; past stem: rft) ('' means empty or null) Suffixes

m rft + m = rftm (person:first; number:singular) (I went)
y rft + y = rfty (person:second; number:singular) (You went)
'' rft = rft (person:third; number:singular) (He/She went)
ym rft + ym = rftym (person:first; number:plural)
yd rft + yd = rftyd (person:second; number:plural)
nd rft + nd = rftnd (person:third; number:Plural)

1.3 Imperative inflection (verb:rftn; PrsentStem:rv ('' means empty or null) Prefix 'b' & Suffixes

'' b + rv = brv (person:second; number:singular)(Go!)
ym b + rv + ym = brvym (person:first; number:plural)(We go or let's go?)
yd b + rv + yd = brvyd (person:second; number:plural)

2. Participle-forming Suffixes

Present participle = present stem + **ndh**

verb: dvydn
present stem: dv
present participle: dvndh (running. e.g. running man)

Past participle = past stem + **h**

verb: frvxtn
past stem: frvxt
past participle: frvxth (sold)

3. Causation Morpheme Infixes & suffix

Causative Present Stem = Present Stem + **|n** or **|ny**

Causative Infinitival = Causative Present Stem + **dn**

Example:

Verb (infinitive)	**Translation**	**Present Stem**	**Causative Present Stem**	**Causative Infinitival**	**Translation**
fhmydn	understand	fhm	fhm|n	fhm|ndn or fhm|nydn	Make understand

Causative Past Stem = Causative Infinitival – **n**

or Causative Past Stem = Causative Present Stem + **d**

Example:

Causative Infinitival	**Translation**	**Causative Past Stem**	**Translation**
fhm|ndn or fhm|nydn	Make understand	Fhm|nd or Fhm|nyd	Made understand

4. Auxiliaries

4.1 Auxiliary bvdn (to be) (AuxBe)

Present Stem = b|^s
Past Stem = bvd

this verb has two series of forms: enclitic and non-enclitic. The enclitic form of this auxiliary (AuxBe) is used in formation of the perfect forms of all verbs. It's inflectional forms are below:

present inflection

|***m***
|***y***
|***st***
|***ym***
|***yd***
|***nd***

4.2 Auxiliary xv|stn (to want) (AuxFuture)

Present Stem = xv|h
Past stem = xv|st

This verb is used as an auxiliary in forming the future tenses. In Shiraz, it's called, "AuxFuture".

4.3 Auxiliary ^sdn (to become) (AuxPassive)

Present Stem = ^sv
Past Stem = ^sd

This auxiliary forms the passive constructions and in Shiraz is called, 'AuxPassive.'

4.4 Auxiliary g^stn (to turn)

Present Stem = grd
Past Stem = g^st

4.5 Auxiliary grdydn (to turn)

Present Stem = grd
Past Stem = grdyd

Both g^stn and grdydn are used as the Passive auxiliaries.

Persian Nonverbal

Suffixes

1. Comparison

Adjective/adverb + **tr** ex. bzrg + tr = bzrgtr (bigger)

2. Superlative

Adjective/adverb + **tryn** ex. bzrgtryn (biggest)

3. Plurals

3. 1. Plural for Nouns
Citation form ($a)
$a + **h|** ex. kt|b + h| = kt|bh| (books)

This morpheme "h|" is most productive and can appear attached or detached.

$a ending in vowel + **y|n** ex. sxngv + y|n = sxngvy|n (speakers)
$a ending in consonant + **yn** ex. m^s|vr + yn = m^s|vryn (consultants)
$a ending in "y" + **vn** ex. mtqy + vn = mtqyvn (religious people)
$a ending in non-vowel + **|n** or **|t** ex. drxt + |n = drxt|n (trees)

$a ending in "h" + **g|n** or **|t** or **J|t**

When attached to a word ending in "h", the "h" is elimintated when forming the plural.
ex. prndh (bird) + g|n = prndg|n (birds)
ex. klmh (word) + |t = klm|t (words)
ex. myvh (fruit) + J|t = myvJ|t (fruits)

3.2. Plural for adjectives and adverbs
Citation form ($b)

$b + **h|** ex. mhrb|n + h| = mhrb|n h|

This morpheme "h|" is most productive and can appear attached or detached.

$b ending in vowel + **y|n** ex. d|n|^sJv + y|n = d|n^sJvy|n (students)
$b ending in consonant + **yn** ex. ms|fr + yn = ms|fryn (travellers)
$b ending in "y" + **vn** ex. |nql|by + vn =|nql|byvn (revolutionaries)

$b ending in consonant + **g|n** or **|t** or **J|t**

When attached to a word ending in "h", the "h" is elimintated when forming the plural.

ex. xbrh + g|n = xbrg|n (specialists)
ex. bndh + g|n = bndg|n (servants)

$b + nonVowel + |**n** or |**t**

ex. mrd + |n = mrd|n (men)

4. Indefinite Morphemes

Appear after the plural on Nouns and Adjectives. Can't appear on Adverbs. If Indefenite was detected, there can not be any ezafe or clitic morphemes; they can be set to null or false.

$b ending in consonant + **y** ex. kt|b + y = kt|by (a book)
$b ending in vowel + **yy** or **iy** ex. b|zJv + yy = b|zJvyy (an interrogator)
$b ending in "y"or"h" + ~|**y** *(~ =space)* ex. xv|nndh + |y = xv|nndh |y (a singer)

Indefinite morpheme can appear following the plural morpheme in which case the translation is "some"

ex. x|nh h|yy or x|nh h|iy (some houses)

5. Enclitic particle

It joins a noun to a relative clause which determines it. This particle is often immediately followed by the relativizer "kh" but it can also be separated from the latter by intervening elements. The form and position of of the enclitic is very similar to that of the Indifinite article (morpheme).
However, different from the Indefinite since the Noun Phrase carrying this enclitic could be interpreted either as a definite or indefinite. The enlclitic attaches to the last element on the Noun Phrase; it can appear on nouns, adjectives, past participles, or classifiers in colloquial speech. It takes the following forms:

($b=NP element)

$b ending in consonant + **y**

ex. kt|by <n> kh rvy myz |st (the book that is on the table)

ex. kt|b xvby <adj> kh … (the good book that …)

$b ending in vowels "v" or "|" + **yy** or **iy**

ex. r|dyvyy kh rvy myz |st (the radio that is on the table)
ex. av|iy ky (the song that …)

$b ending in silent "h" or in final form of "y" + |**y**

ex. bndh |y kh mybynyd (the servant that you see)

Note: since the surface form of the Indefinite and Enclitic morphemes are identical, There is the "indefiniteEnclitic" feature in Shiraz. Since only the structural position of the nominal element on which these morphemes appear can disambiguate them, they were supposed to be disambiguated at the level of syntactic analysis.

5. Ezafe

Ezafe links the elements within a nominal phrase. Is used to relate a lexical element to its modifiers or to the possessors. It can also appear on Prepositions linking them to their noun phrase element.

Ezafe appears on Nouns, Infinitives, Adjectives, Adverbs and quantifiers. This vowel is usually not written. But in certain cases (see below) it is written as "y". If

$b ending in Vowel "v" or "|" + **y**

ex. Sd| + y = Sd|y (sound)
ex. p| + y = p|y (foot)

together: Sd|y p|y mn (Sound of my foot)

$b ending in "h" + **;** (called 'hamze')

ex.x|nh + ; = x|nh ; Hassan (Hassan's house)

Hamze is optional and usually doesn't appear in written text.

When the ezafe is detected, there can be no indefinite or enclitic or clitic morpheme; they can be set to null or false.

For ezafe, if word ends in a nonvowel and the ezafe value can't be determined with certainty (since it's not written), it is then set as Undefined.

6. Clitics (Personal Suffixes)

6.1 Possessive Clitics

The possessive suffixes attach to the last element of the Noun Phrase, hence, they appear on nouns (kt|bm), adjectives (kt|b xvbm), infinitivals (bvdnm:my being), past participles (m|^syn t@myr ^sdh |m) and classifiers(?). In addition they can appear on an adjectival that is being used as a definite noun. (e.g. sfyd^s (the white one)).

a) for words ending in consonant or vowel + **y**

m kt|b + m = kt|bm (my book) (person:first, number:singular)
t kt|b + t = kt|bt (your book) (person:second, number:singular)
^s kt|b + ^s= kt|b^s (his or her book) (person:third, number singular)
m|n kt|b + m|n = kt|bm|n (our book) (person:first, number: plural)

t|n kt|b + t|n = kt|bt|n (your book) (person:first, number: plural)
^s|n kt|b + ^s|n = kt|b^s|n (their book) (person:first, number: plural)

b) for words ending in vowels, | or **v**

ym	d|rv + ym = d|rvym	(my medication)(person:first, number:singular)
yt	d|rv + yt = d|rvyt	(your medication)(person:second, number:singular)
y^s	d|rv + y^s = d|rvy^s	(their medication)(person:third, number singular)
ym|n	d|rv + ym|n = d|rvym|n	(our medication)(person:first, number: plural)
t|n	d|rv + yt|n = d|rvyt|n	(your medication)(person:first, number: plural)
^s|n	d|rv + y^s|n = d|rvy^s|n	(their medication)(person:first, number: plural)

c) for words ending in final form **h** or **y**

#all below are preceded by ~ (space)

|m	nv^s|bh + |m = nv^s|bh |m	(my drink)
|t	nv^s|bh + |t = nv^s|bh |t	(your drink)
|^s	nv^s|bh + |^s= nv^s|bh |^s	(his or her drink)
m|n	nv^s|bh + m|n = nv^s|bh m|n	(our drink)
t|n	nv^s|bh + t|n = nv^s|bh t|n	(your drink)
^s|n	nv^s|bh + ^s|n = nv^s|bh ^s|n	(their drink)

6.2 Object Clitics

Object clitics, <u>mostly colloquial</u>, are accusative forms of the personal pronouns. These clitics can be attached to prepositions and to transitive infinitivals:

Transitive Infinitival
zdn (v) + ^s|n: (hitting + Obj.3pl)

ex. zdn^s|n k|r drsty nyst (hitting them is not a good thing to do)

tnbyh krdn(lv) + t|n: (punishing + Obj.2pl)

ex. tnbyh krdnt|n xvb nyst (punishing you is not appropriate)

Preposition
Jlv (pre) + ym (front + Obj.1s) = Jlvym (in front of me)

ex. Jlvym m|^syn r| t|my krd (He/she fixed the car in front of me)

Transitive verbs or light verbs
dyd + m + t (saw + 1st + Obj.2s) dydmt (I saw you)

Compound Transitive Verbs:
br +^s d|^st (up + Obj.3s took) = br^sd|^st (He/She picked it up)

[or clitic on end of compound verb: *colloquial*
brd|^st + ^s = brd|^st^s (He/She picked it up)]

Compound Transitive Verb:
Clitic on preverbal nominal element:
dvst + t d|r + m (like + Obj.2s have+1s) = dvstt d|rm (I like you)

[or Clitic on end of compound verb: *colloquial*
dvst d|r+m+t (like + have + 1st + Obj.2s) = dvst d|rmt (I like you)

If no ezafe, IndefiniteEnclitic or Clitic are found on nouns, adjectives, adverbs, prepositions, we can set the values for all these features to negative.

7. Ordinals

These morphemes are used to form ordinal numbers from cardinal ones.
m dv (number) + m = second

The ordinals formed behave as adjectives and adverbs. They indicate numerical rank.

my dv + my = dvmy (the second car is …)
Used in colloquial speech and for lower numbers; behave as adjectives

myn dv + myn = dvmyn (the second book, the third book, …)
These ordinals indicate the unity which completes a series; specially used to designate an anniversary. They behave like an adjective and can NOT be used as adverbs.

Irregular ordinals are specified in the dictionary (nxstyn).

8. Copula (not covered in Shiraz)

Copula can appear on nouns, propernames, adjectives, past participles and classifiers. Not covered in Shiraz because it is a verbal element that appears on nominal constituents, and required to be separated from the nominal feature structure (Unification-based Persian Morphology page 15)

a. suffixes that appear following an element ending in consonant
m xvb(adj) + m = xvbm (person: first; number singular)(I am good)
y xvv + y = xvby (person: second;number: singular)
st or |**st** xvb + st = xvbst (person: third;number: singular)
ym xvb + ym = xvbym (person: first; number: plural)
yd xvb + yd = xvbyd (person: second; number:plural)
nd xvb + nd = xvbnd (person: third; number: plural)

b. suffixes that appear after element ending in vowels "|" or "v"
ym zyb|(adj) +ym = zyb|ym (person:first;number singular)(I'm beautiful)

yy zyb| + yy = zyb|yy (person: second;number: singular)(you're …)
st or **|st** zyb| + st = zyb|st (person: third;number: singular)
yym zyb| + yym = zyb|yym (person: first; number: plural)
yyd zyb| + yyd = zyb|yyd (person: second; number:plural)
ynd zyb| + ynd = zyb| (person: third; number: plural)

c. suffixes that appear following certain words ending in silent 'h' and 'y' (in final form only)

|m xv|bydh + |m = xv|bydh |m (person: first;number singular)(I'm asleep)
|y xv|bydh + |y = xv|bydh |y (person: second;number: singular)
|st xv|bydh + |st = xv|bydh |st (person: third;number: singular)
|ym xv|bydh + |ym = xv|bydh |ym (person: first; number: plural)
|yd xv|bydh + |yd = xv|bydh |yd (person: second; number:plural)
|nd xv|bydh + |nd = xv|bydh |nd (person: third; number: plural)

Persian Conjugation

Active Voice

Simple Form

Present Indicative = **my +** Present Stem + Present Inflection

Present Subjunctive = **b** + Present Stem + Present Inflection

Imparative = **b** + Present Stem + Imparative Inflection

Preterite = Past Stem + Past Inflection

Imperfect **=** **my** + PastStem + PastInflection

Compound Forms

Perfect **=** PastParticiple + AuxBe[Present]

CompoundImperfect = '**my**' + PastParticiple + AuxBe[Present]

Pluperfect = PastParticiple + AuxBe[Preterite]
AuxBe[Preterite] = AuxBe[PastStem] + PastInflection

DoubleCompound **=** PastParticiple + AuxBe[Perfect]
AuxBe[Perfect] = AuxBe[PastParticiple] + AuxBe[Present]

CompoundSubjunctive **=** PastParticiple + AuxBe[Subjunctive]
AuxBe[Subjunctive] = AuxBe[PresentStem] + PresentInflection

Future **=** AuxFuture + PastStem

Passive Voice

Indicative

Present **=** PastParticiple + AuxPassive[Present]
frvxth my^svd [forukhte mishavad] (is being sold)

Preterite **=** PastParticiple + AuxPassive[Preterite]
frvxth ^sd [forukhte shod] (was sold)

Imperfect **=** PastParticiple + AuxPassive[Imperfect]
frvxth my^sd [forukhte mishod] (was being sold)

Perfect = PastParticiple + AuxPassive[Perfect]
frvxth ^sdh |st [forukhte shode ast] (has been sold)

CompoundImperfect = PastParticiple + AuxPassive[CompoundImperfect]
frvxth my^sdh |st [forukhte mishode ast] (was being sold)

Pluperfect= PastParticiple + AuxPassive[Pluperfect]
frvxth ^sdh bvd [forukhte shode bud] (had been sold)

DoubleCompound = PastParticiple + AuxPassive[DoubleCompound]
frvxth ^sdh bvdh |st [forukhte shode budh ast] (had been sold)

Future = PastParticiple + AuxPassive[Future]
frvxth xv|hd ^sd [forukhte khAhad shod] (will be sold)

Subjunctive

Subjunctive = PastParticiple + AuxPassive[subjunctive]
frvxth ^svd [forukhte shavad] (that he/she/it be sold)

CompoundSubjunctive = PastParticiple + AuxPassive[CompoundSubjunctive]
frvxth ^sdh b|^sd [forukhte shode bud] (that he/she/it has been sold)

Imperative

Imperative = PastParticiple + AuxPassive[imperative]
frvxth ^sv [forukhte sho] (be sold)

Irregular Verbs

present form of "bvdn"

hstm [hastam] (I am, I exist)
hsty [hasti] (you are, you exist)
hst [hast] (he/she/it is, he/she/it exists)
hstym [hastim] (we are, we exist)
hstyd [hastid] (you are, you exist)
hstnd [hastand] (they are, they exist)

present form of "bvdn" - negative

nystm [nistam] (I am not)
nysty [nisti] (you are not)
nyst [nist] (he/she/it is not)
nystym [nistim] (we are not)
nystyd [nistid] (you are not)
nystnd [nistand] (they are not)

present form of "d|^stn"

d|rm [dAram] (I have)
d|ry [dAri] (you have)
d|rd [dArad] (he/she/it has)

d|rym [dArim] (we have)
d|ryd [dArid] (you have)
d|rnd [dArand] (they have)

Negation (negation prefix "n")

Active voice

In the active voice, negative morpheme on beginning of simple verbs. In the compound forms, the negative prefix also appears on the beginning of the conjugated verbal element since it attaches to the past participle of the main verb. The only exception is the Double Compound Past**, in which the negative attaches to the past participle form of the auxiliary and not on the main verb. In addition, if a modal is present in the sentence, the negation can appear on the modal element. The Present Participle does not carry negation since it is often used as a nominal element. The negative affix can appear on Past Participles and Infinitivals.

In the Passive voice, with the exception of the Future Passive, the negative prefix always appears on the passive auxiliary. Examples are given below for each tense in the Active Voice. The negative prefix can be seen on the beginning of all the verb forms with the exception of the Double Compound Past. In this case, the *n* morpheme is attached to the past participle of the auxiliary *bvdn*[*budan*] (to be). The negative prefix has the form *n* before consonants and the form *ny* before the vowels | *[A]* and *v [u]*.

Active Voice:

Past Participle *nfrvxth [naforukhte] (not sold)*

Present Indicative *nmyfrv^sm [nemiforusham] (I don't sell/ I am not selling)*

Preterite *nfrvxtm [naforukhtam] (I didn't sell)*

Imperfect *nmyfrvxtm [nemiforukhtam] (I wasn't selling/I didn't use to sell)*

Perfect *nfrvxth~|m [naforukhteam] (I have not sold)*

Compound Imperfect *nmyfrvxth~|m [nemiforukhteam] (I have not been selling)*

Pluperfect *nfrvxth bvdm [naforukhte budam] (I had not sold)*

Future *nxv|hm frvxt [nakhAham forukht] (I will not sell)*

Present Subjunctive *nfrv^sm [naforusham] (that I do not sell)*

Compound Subjunctive *nfrvxth b|^sm [naforukhte bAsham] (that I have not sold)*

Imperative *nfrv^s [naforush] (do not sell)*

Double Compound Past** *frvxth nbvdh~|m [forukhte nabudeam] (I had not sold)*

The Passive forms are listed below. In all tenses, the negative prefix attaches to the passive auxiliary ^*sdn*[*shodan*] (to become) and never on the verb. The only exception is the Future tense in the passive; in this case, the negative prefix attaches to the future auxiliary.

Passive Voice:

Present Indicative *frvxth nmy^svd [forukhte nemishavad] (is not being sold)*

Preterite *frvxth n^sd [forukhte nashod] (was not sold)*

Imperfect *frvxth nmy^sd [forukhte nemishod] (wasn't being sold)*

Perfect *frvxth n^sdh |st [forukhte nashode ast] (has not been sold)*

CompoundImperfect *frvxth nmy^sdh |st[forukhth nemishode ast] (was not being sold)*

Pluperfect *frvxth n^sdh bvd [forukhte nashode bud](had not been sold)*

DoubleCompound *frvxth n^sdh budh |st [forukhte nashode bude ast] (had not been sold)*

Present Subjunctive *frvxth n^svd [forukhte nashavad] (that it not be sold)*

Compound Subjunctive *frvxth n^sdh b|^sd [forukhte nashode bAshad] (that it has not been sold)*

Imperative *frvxth n^sv [forukhte nasho] (do not be sold)*

Future *frvxth nxv|hd ^sd [forukhte nakhAhad shod] (will not be sold)*