CSE528

Natural Language Processing

Venue:ADB-405 SLOTS: A2+TA2 Topic: Text Processing

Prof. Tulasi Prasad Sariki,

SCSE, VIT Chennai Campus

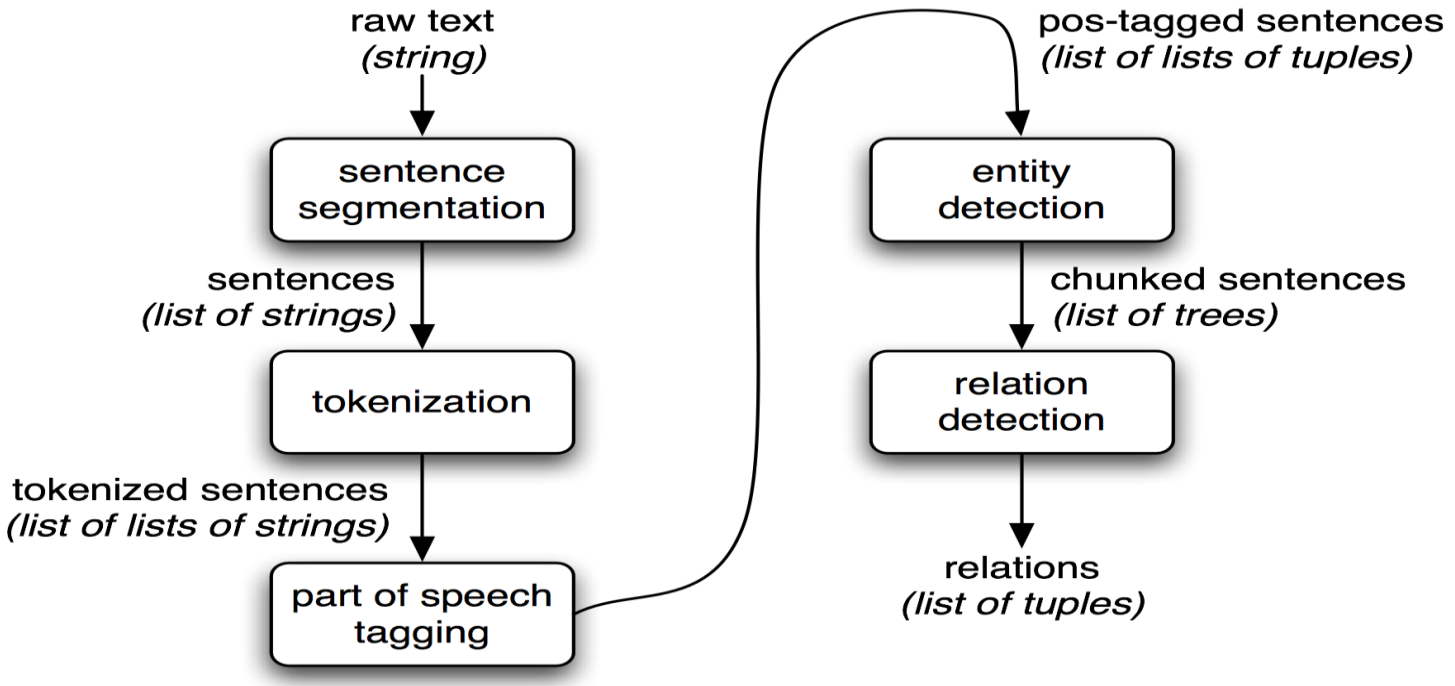
www.learnersdesk.weebly.com

Contents

- Text Processing
- Text Preprocessing
- Challenges in Text Preprocessing
- Types of Writing Systems

Text Processing

In the linguistic analysis of a digital natural language text, it is necessary to clearly define the characters, words, and sentences in any document.



Text Preprocessing

The task of converting a raw text file, essentially a sequence of digital bits, into a well-defined sequence of linguistically meaningful units:

- at the lowest level characters representing the individual graphemes in a language's written system,
- Words consisting of one or more characters,
- sentences consisting of one or more words

Text preprocessing is an essential part of any NLP system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages.

Text / Word segmentation

Text segmentation is the process of converting a well-defined text corpus into its component words and sentences.

This is very important task to work on morphology and syntax levels of NLP.

Word segmentation breaks up the sequence of characters in a text by locating the word boundaries, the points where one word ends and another begins.

For computational linguistics purposes, the words thus identified are frequently referred to as tokens, and word segmentation is also known as tokenization.

Sentence Segmentation / Text Normalization

Sentence segmentation is the process of identifying sentence boundaries between words in different sentences.

Since most written languages have punctuation marks that occur at sentence boundaries, sentence segmentation is frequently referred to as sentence boundary detection, sentence boundary disambiguation

Text normalization is a related step that involves merging different written forms of a token into a canonical normalized form; for example, a document may contain the equivalent tokens “Mr.”, “Mr”, “mister”, and “Mister” that would all be normalized to a single form.

Challenges of Text Preprocessing

The type of writing system (SCRIPT) used for a language is the most important factor for determining the best approach to text preprocessing.

It needs:

- at least one set of defined base elements or symbols, individually termed *characters* and collectively called a **script**;
- at least one set of rules and conventions (orthography) understood and shared by a community, which arbitrarily assigns meaning to the base elements (graphemes), their ordering and relations to one another;
- at least one language (generally spoken) whose constructions are represented and able to be recalled by the interpretation of these elements and rules

Classification of Systems

Classification by Daniels

Type	Each symbol represents	Example
Logographic	morpheme	Chinese characters
Syllabic	syllable or mora	Japanese kana
Alphabetic	phoneme (consonant or vowel)	Latin alphabet
Abugida	phoneme (consonant+vowel)	Indian Devanāgarī
Abjad	phoneme (consonant)	Arabic alphabet
Featural	phonetic feature	Korean hangul

Logographic writing systems

In a logographic writing system, in theory, each symbol (word or morpheme) represents one idea example: Chinese

Friendship

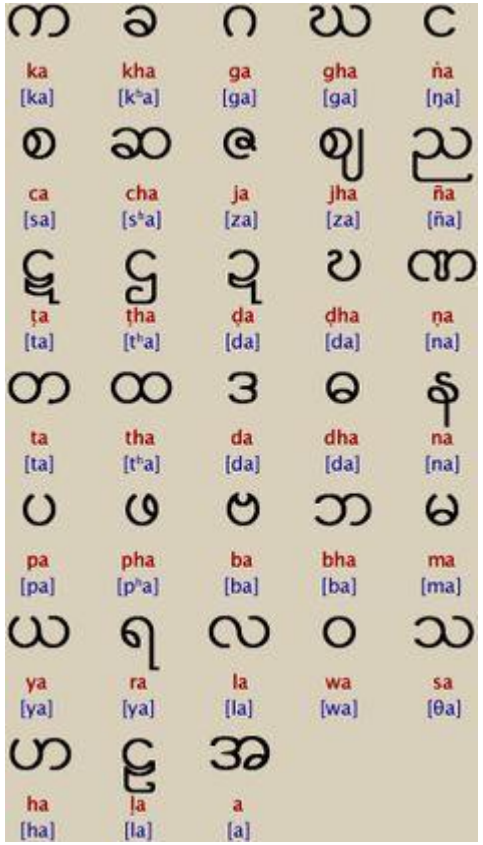
Happiness

Fortune

Prosperity

Logophonetic Writing Systems

Definition: there are two major types of signs, ones denoting morphemes and ones denoting sounds. (ex) Egyptian, Japanese and sumerian



Abugida

South Asian scripts such as Brahmi and its descendants fit into both syllabary and alphabet.

It is syllabic because the basic sign contains a consonant and a vowel.

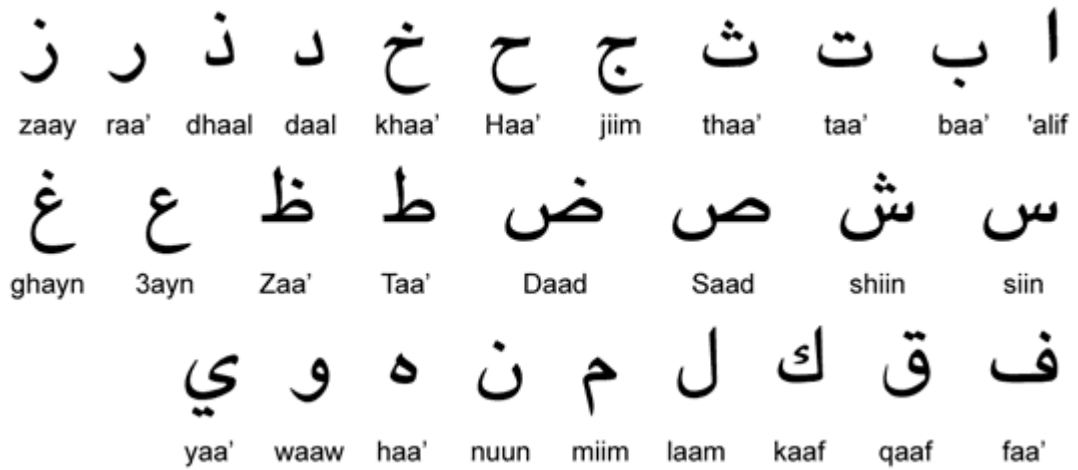
Greek had CV, CVC, CCVC, CVCC syllable structures, so they invent a way to cut down syllables to consonant and vowels

Alphabetic

A system of consonant and vowel symbols that, either individually or in combinations, represent the speech sounds of a written language (ex) English

Abjad or Consonantal Alphabet

alphabetic writing systems in which only the consonants in words are written, and the vowels are left out (ex) Hebrew, Arabic



Character representation

How Characters in languages can be represented?

At its lowest level, a computer-based text or document is merely a sequence of digital bits in a file.

The first essential task is to interpret these bits as characters of a writing system of a natural language.

Unicode

Fundamentally, computers just deal with numbers. They store letters and other characters by assigning a number for each one.

Before Unicode was invented, there were hundreds of different encoding systems for assigning these numbers.

Even for a single language like English no single encoding was adequate for all the letters, punctuation, and technical symbols in common use.

Unicode

These encoding systems also conflict with one another. That is, two encodings can use the same number for two *different* characters, or use different numbers for the *same* character.

Unicode covers all the characters for all the writing systems of the world, modern and ancient. It also includes technical symbols, punctuations, and many other characters used in writing text. The Unicode Standard is intended to support the needs of all types of users, whether in business or academia, using mainstream or minority scripts.

Types of Encoding

Two Types Encoding

Character Encoding

- ASCII, ISCII, Unicode

Font Encoding

- Eenadu, vaartha, Kumudam , Daily Thanthi

ASCII Features

American Standard Code for Information Interchange

7-bit code

8^{th} bit is unused (or used for a parity bit)

$2^7 = 128$ codes

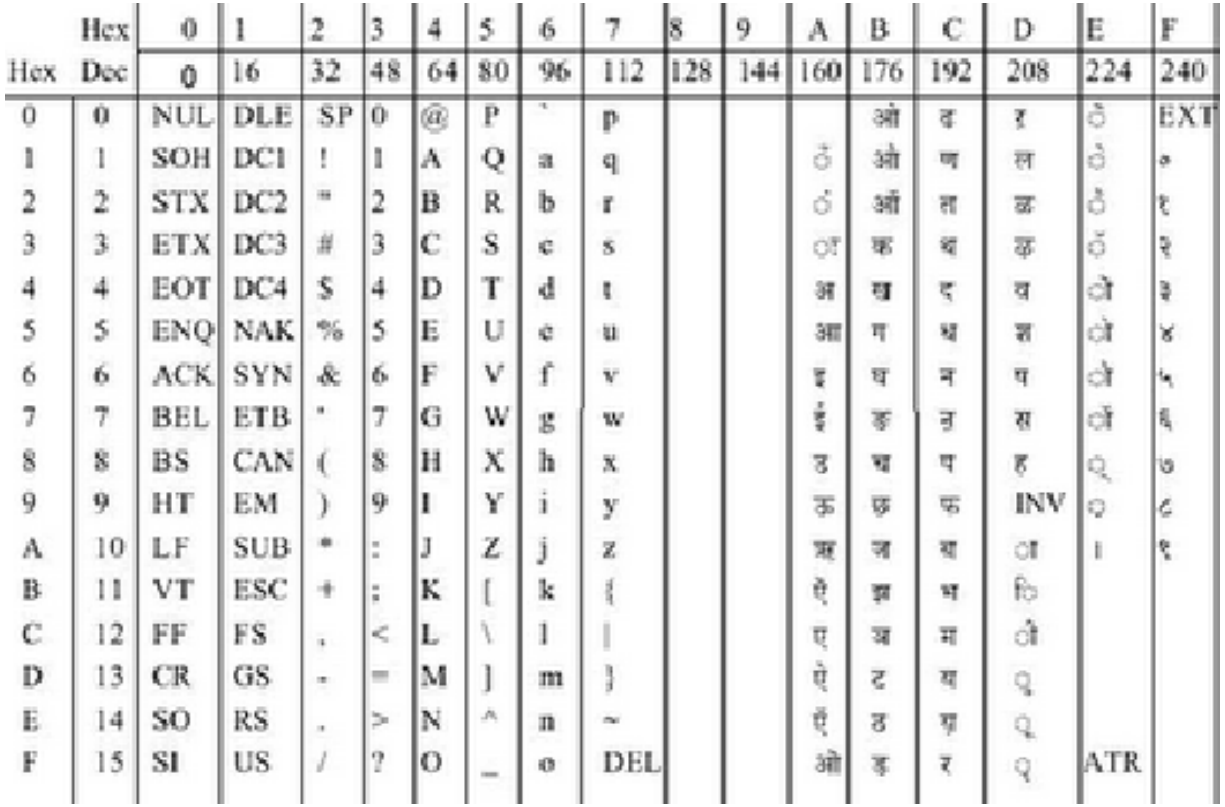
Two general types of codes:

- 95 are “Graphic” codes (displayable on a console)
- 33 are “Control” codes (control features of the console or communications channel)

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

ISCII(Indian Standard Code for Information Interchange)



Hex	Dec	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Dec	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
0	0	NUL	DLE	SP	0	@	P	`	p				ओ	द	ह	ॅ	EXT
1	1	SOH	DC1	!	1	A	Q	a	q			ं	ओ	न	ल	े	०
2	2	STX	DC2	"	2	B	R	b	r			ः	औ	प	ळ	ै	१
3	3	ETX	DC3	#	3	C	S	c	s			ऀ	क	फ	क्ष	ॉ	२
4	4	EOT	DC4	$	4	D	T	d	t			अ	ख	ब	ज्ञ	ो	३
5	5	ENQ	NAK	%	5	E	U	e	u			आ	ग	भ	श	ौ	४
6	6	ACK	SYN	&	6	F	V	f	v			इ	घ	म	ष	ौ	५
7	7	BEL	ETB	'	7	G	W	g	w			ई	ङ	य	स	ौ	६
8	8	BS	CAN	(	8	H	X	h	x			उ	च	र	ह	़	७
9	9	HT	EM	)	9	I	Y	i	y			ऊ	छ	ऱ	INV	़	८
A	10	LF	SUB	*	:	J	Z	j	z			ऋ	ज	ल	ा	।	९
B	11	VT	ESC	+	;	K	[	k	{			ऌ	झ	ळ	ि		
C	12	FF	FS	,	<	L	\	l	|		ऍ	ञ	ऴ	ी		
D	13	CR	GS	-	=	M	]	m	}			ऎ	ट	व	ु		
E	14	SO	RS	.	>	N	^	n	~			ए	ठ	श	ू		
F	15	SI	US	/	?	O	_	o	DEL			ऐ	ड	ष	ृ	ATR	

It is a coding scheme for representing various writing systems of India. It encodes the main Indic scripts and a Roman transliteration.

The supported scripts are: Assamese, Bengali (Bengla), Devanagari, Gujarati, Gurmukhi, Kannada, Malayalam, Oriya, Tamil, and Telugu.

One motivation for the use of a single encoding is the idea that it will allow easy transliteration from one writing system to another.

Unicode

Unicode is a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems.

The latest version (*Unicode 7.0*) of Unicode contains a collection of more than 110,000 characters covering 100 scripts and various symbols.

Unicode can be implemented by different character encodings. The most commonly used encodings are UTF-8, UTF-16.

http://www.unicodetables.com/

http://www.unicode.org/

Structural Differences with ISCII

Unicode is stateless:

- No shifting to get different scripts
- Each character has a unique number

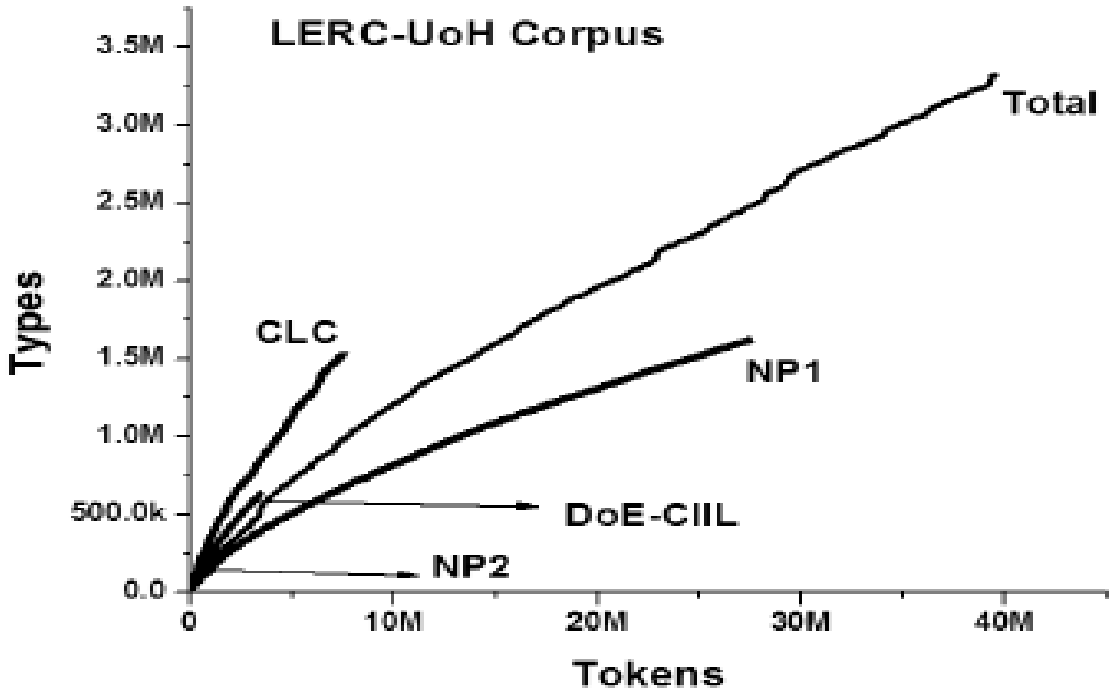
Unicode is uniform:

- No extension bytes necessary
- All characters coded in the same space

Yudit is a free Unicode plain-text editor for Unix-like systems.

http://www.yudit.org/

Type vs Token



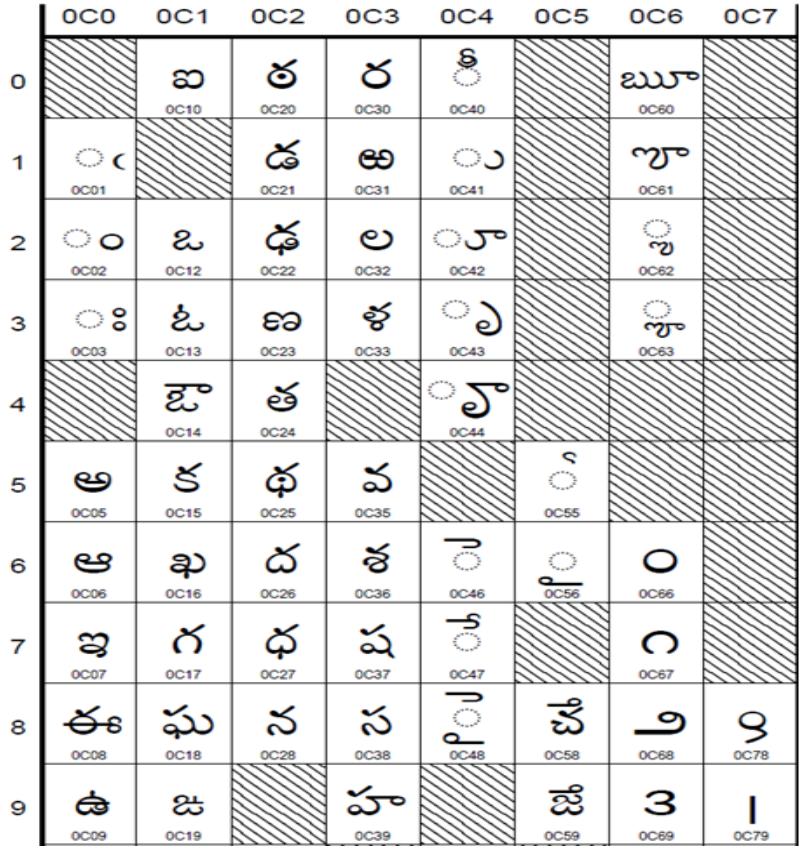
Example:

Consider the sentence below
A rose is a rose is a rose

There are three word types in the sentence: "rose", "is" & "a".

There are eight word tokens

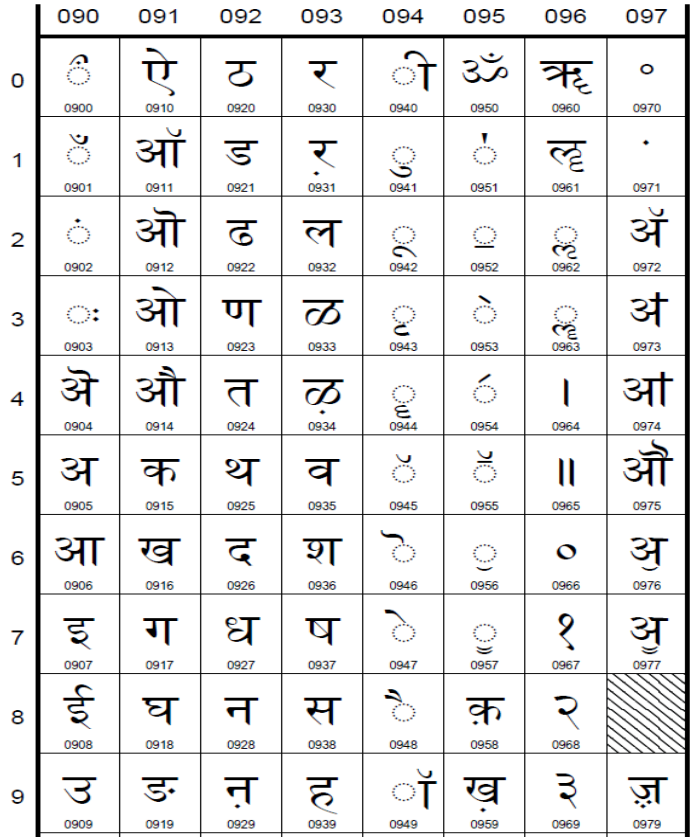
Telugu



	0C0	0C1	0C2	0C3	0C4	0C5	0C6	0C7
0		ఐ 0C10	ఠ 0C20	ర 0C30	◌ీ 0C40		ౠ 0C60	
1	◌ఁ 0C01		డ 0C21	ఱ 0C31	◌ు 0C41		ౡ 0C61	
2	◌ం 0C02	ఒ 0C12	ఢ 0C22	ల 0C32	◌ూ 0C42		◌ౢ 0C62	
3	◌ః 0C03	ఓ 0C13	ణ 0C23	ళ 0C33	◌ృ 0C43		◌ౣ 0C63	
4		ఔ 0C14	త 0C24		◌ౄ 0C44			
5	అ 0C05	క 0C15	థ 0C25	వ 0C35		◌ౕ 0C55		
6	ఆ 0C06	ఖ 0C16	ద 0C26	శ 0C36	◌ె 0C46	◌ౖ 0C56	౦ 0C66	
7	ఇ 0C07	గ 0C17	ధ 0C27	ష 0C37	◌ే 0C47		౧ 0C67	
8	ఈ 0C08	ఘ 0C18	న 0C28	స 0C38	◌ై 0C48	ౘ 0C58	౨ 0C68	౸ 0C78
9	ఉ 0C09	ఙ 0C19		హ 0C39		ౙ 0C59	౩ 0C69	౹ 0C79

A	ఊ 0C0A	చ 0C1A	ప 0C2A		◌ొ 0C4A		౪ 0C6A	౺ 0C7A
B	ఋ 0C0B	ఛ 0C1B	ఫ 0C2B		◌ో 0C4B		౫ 0C6B	౻ 0C7B
C	ఌ 0C0C	జ 0C1C	బ 0C2C		◌ౌ 0C4C		౬ 0C6C	౼ 0C7C
D		ఝ 0C1D	భ 0C2D	ఽ 0C3D	◌్ 0C4D		౭ 0C6D	౽ 0C7D
E	ఎ 0C0E	ఞ 0C1E	మ 0C2E	◌ా 0C3E			౮ 0C6E	౾ 0C7E
F	ఏ 0C0F	ట 0C1F	య 0C2F	◌ి 0C3F			౯ 0C6F	౿ 0C7F

Devanagari



	090	091	092	093	094	095	096	097
0	◌ 0900	ऐ 0910	ठ 0920	र 0930	◌ी 0940	◌ॕ 0950	ॠ 0960	॰ 0970
1	◌ँ 0901	ऑ 0911	ड 0921	ऱ 0931	◌ु 0941	◌ॖ 0951	ॡ 0961	ॱ 0971
2	◌ं 0902	ओ 0912	ढ 0922	ल 0932	◌ू 0942	◌ॗ 0952	◌ॢ 0962	ॲ 0972
3	◌ः 0903	ओ 0913	ण 0923	ळ 0933	◌ृ 0943	◌॓ 0953	◌ॣ 0963	ॳ 0973
4	ऄ 0904	औ 0914	त 0924	ऴ 0934	◌ॄ 0944	◌॔ 0954	। 0964	ॴ 0974
5	अ 0905	क 0915	थ 0925	व 0935	◌ॅ 0945	◌ॕ 0955	॥ 0965	ॵ 0975
6	आ 0906	ख 0916	द 0926	श 0936	◌ॆ 0946	◌ॖ 0956	० 0966	ॶ 0976
7	इ 0907	ग 0917	ध 0927	ष 0937	◌े 0947	◌ॗ 0957	१ 0967	ॷ 0977
8	ई 0908	घ 0918	न 0928	स 0938	◌ै 0948	क़ 0958	२ 0968	
9	उ 0909	ङ 0919	ऩ 0929	ह 0939	◌ॉ 0949	ख़ 0959	३ 0969	ज़ 0979

A	ऊ 090A	च 091A	प 092A	◌ॊ 093A	◌ो 094A	ग़ 095A	४ 096A	य 097A
B	ऋ 090B	छ 091B	फ 092B	◌ा 093B	◌ो 094B	ज़ 095B	५ 096B	ग्र 097B
C	ऌ 090C	ज 091C	ब 092C	◌़ 093C	◌ौ 094C	ड़ 095C	६ 096C	ज्र 097C
D	ऍ 090D	झ 091D	भ 092D	ऽ 093D	◌् 094D	ढ़ 095D	७ 096D	? 097D
E	ऎ 090E	ञ 091E	म 092E	◌ा 093E	◌ॎ 094E	फ़ 095E	८ 096E	ड्र 097E
F	ए 090F	ट 091F	य 092F	◌ि 093F	◌ॏ 094F	य़ 095F	९ 096F	ब्र 097F

Font

A *font* file is a binary file that contains glyphs, or “pictures”, of symbols representing the building blocks of a displayable character set.

Depending on the language, multiple glyphs can comprise a single character.

Code Table

In basic terms, a *code table* is a two column list that maps a numerical value to a glyph. The most widely used code table is Unicode

Font

Encoding

Encoding values are “stored” from a code table. There are many different encoding types to choose from depending on the application.

- UTF-8
- UTF-16 (UCS(universal character set)-2)
- UTF-32 (UCS(universal character set)-4)

Allows us to generate displays of text strings in many different languages by using fonts which contain the glyphs corresponding to their alphabet

The computer system takes each code and displays the glyph associated with it which is displayed on a monitor or printed out.

Font

The glyphs may be viewed as the building blocks for the letter to be displayed where, by placing the glyphs one after another, the required display is generated.

Fonts also incorporate a feature whereby some of the glyphs may be defined to have zero width even though they extend over a horizontal range

Thus when the system places a zero width glyph next to another, the two are superimposed and thus permit more complex shapes to be generated, such as accented letters.

Sentence

A **sentence** is a group of words that are put together to mean something.

A sentence is the basic unit of language which expresses a complete thought.

It does this by following the grammatical rules of syntax.

Sentence Boundary Disambiguation

- People use . ? and !
- Sometimes ;
- End-of-sentence marks are overloaded.

Sentence Boundary Disambiguation

English employs whitespace between most words and punctuation marks at sentence boundaries, but neither feature is sufficient to segment the text completely and unambiguously.

Tibetan and Vietnamese both explicitly mark syllable boundaries, either through layout or by punctuation, but neither marks word boundaries.

Written Chinese and Japanese have adopted punctuation marks for sentence boundaries, but neither denotes word boundaries.

Period - most ambiguous. Decimals, e-mail addresses, abbreviations, initials in names, honorific titles.

Sentence Boundary Disambiguation

For example:

U.S. Dist. Judge Charles L. Powell denie motions made by defense attorneys Monday in Portland's insurance fraud trial. Of the handful of painters that Austria has produced in the 20th century, only one, Oskar Kokoschka, is widely known in U.S. This state of unawareness may not last much longer.

Sentence boundary detection by humans is tedious, slow, error-prone, and extremely difficult to codify.

Algorithmic syntactic sentence boundary detection is a necessity.

POS tagging and syntax can be done on sentences

Related Work

As of 1997:

"identifying sentences has not received as much attention as it deserves." [Reynar and Ratnaparkhi1997]

"Although sentence boundary disambiguation is essential . . ., it is rarely addressed in the literature and there are few public-domain programs for performing the segmentation task." [Palmer and Hearst1997]

Two approaches

- Rule based approach
- Machine-learning-based approach

Related Work

Rule based

- Regular expressions
  - [Cutting1991]
  - Mark Wasson converted grammar into a finite automata with 1419 states and 18002 transitions.
- Lexical endings of words
  - [Müller1980] uses a large word list.

Machine-learning-based approach

- [Riley1989] uses regression trees.
- [Palmer and Hearst1997] uses decision trees or neural network.

Maximum Entropy Approach

Potential sentence boundaries are identified by scanning the text for sequences of characters separated by whitespace (tokens) containing one of the symbols !, . or ?.

The system that focused on maximizing performance used the following hints, or contextual "templates":

The Prefix, The Suffix

The presence of particular characters in the Prefix or Suffix

Whether the Candidate is an honorific (e.g. *Ms., Dr., Prof.)*

Maximum Entropy Approach

Whether the Candidate is a corporate designator (e.g. Corp., *M.L.A., M.L.C.)*

Features of the word left of the Candidate

Features of the word right of the Candidate

The templates specify only the form of the information. The exact information used by the maximum entropy model for the potential sentence boundary marked by Corp. in Example sentence would be:

- ANLP Corp. chairman Dr. Smith resigned.
- PreviousWordIsCapitalized, Prefix=Corp, Suffix=NULL, PrefixFeature=CorporateDesignator.

Maximum Entropy Approach

For each potential sentence boundary token (., ?, and !), we estimate a joint probability distribution p of the token and its surrounding context, both of which are denoted by c, occurring as an actual sentence boundary.

The distribution is given by:

p(b,c) $= \pi \prod_{j=1}^{k} \alpha_j^{f_j(b,c)}$

Where b∈ {no, yes}, where the α_j's are the unknown parameters of the model, and where each α_j corresponds to a f_j, or a *feature.*

Thus the probability of seeing an actual sentence boundary in the context c is given by p(yes, c).

Corpus

Corpus is a large collection of text covering different domains, styles, territorial and social variants of usage etc.

A *corpus* is a collection of pieces of language that are selected and ordered according to explicit linguistic criteria in order to be used as a sample of the language.

A corpus provides grammarians, lexicographers, and others a better description of a language.

Chomsky's Critique of Corpus-Based Methods

1. Corpora model performance, while linguistics is aimed at the explanation of competence

   If you define linguistics that way, linguistic theories will never be able to deal with actual, messy data

2. Natural language is in principle infinite, whereas corpora are finite, so many examples will be missed

   Excellent point, which needs to be understood by anyone working with a corpus.

   But does that mean corpora are useless?

   Introspection is unreliable (prone to performance factors), and pretty useless with small and unrepresentative data.

   Insights from a corpus might lead to generalization/induction beyond the corpus– if the corpus is a good sample of the "text population"

3. Ungrammatical examples won't be available in a corpus

   Depends on the corpus, e.g., spontaneous speech, language learners, etc.

Corpus

Corpora analysis provide lexical information, morpho-syntactic information, syntactic as well as semantic information.

Variety of Corpus

- Raw corpus
- POS tagged
- Parsed
- Multilingual aligned
- Spoken language
- Semantic tagged

Corpus

Raw Corpus

The texts are segmented into sentences and paragraphs

- Reuters corpus (180 Million Word)
- CIIL corpus (3 Million words for 10 major Indian languages)

POS Tagged Corpus

texts in corpus are annotated with Part Of Speech tags information

- BNC tagged corpus(100 Million CLAWS tagset)

Corpus

Parsed

Each sentence is annotated with a phrase-structure parse marking the boundaries of sentence, clause, phrase and coordinated word constituents.

- Lancaster Parsed Corpus (British English)
- Susanne parsed corpus

Semantic Corpus

Sense tagged corpus

- The FrameNet lexical database contains around 1,200 semantic *frames*, 13,000 *lexical units* (a pairing of a word with a meaning; polysemous words are represented by several *lexical units*) and over 190,000 example sentences

Corpus

Multilingual aligned

Identification of the corresponding sentences in multiple languages and align them

- CRATER**:-** Multilingual Aligned Annotated Corpus (English, French, Spanish)
- JRC-Acquis Multilingual Parallel Corpus: collection of parallel texts in the following **22 languages**: Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish etc.
- Parallel corpora are useful for all types of cross-lingual research

Uses of Corpora

Lexicography / terminology

Linguistics / computational linguistics

Dictionaries & grammars (Collins Cobuild) English Dictionary for Advanced Learners; Longman Grammar of Spoken and Written English

Critical Discourse Analysis

- Study texts in social context

- Analyze texts to show underlying ideological meanings and assumptions

- Analyze texts to show how other meanings and ways of talking could have been used….and therefore the ideological implications of the ways that things were stated

Uses of Corpora

Literary studies

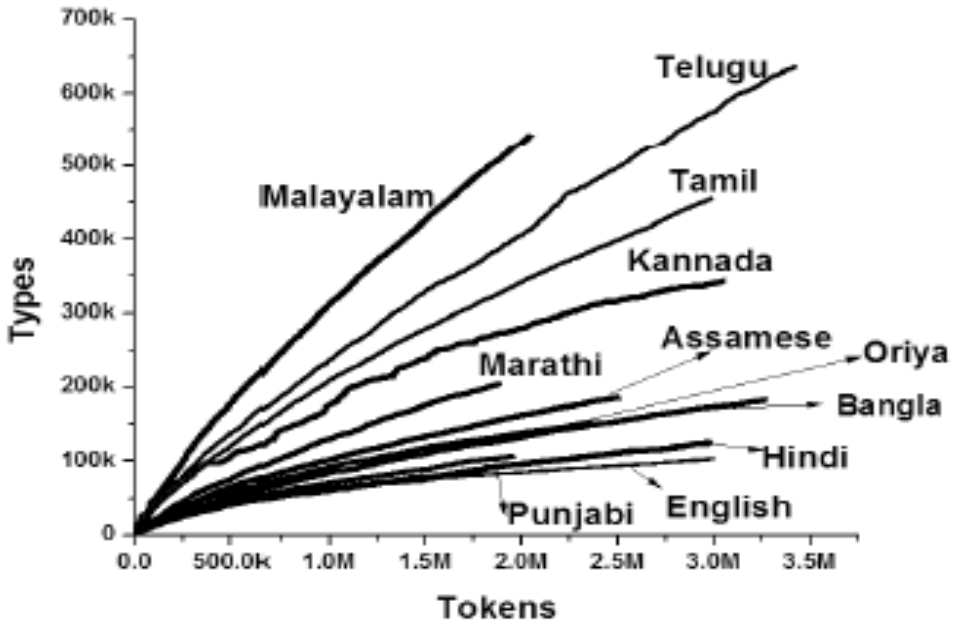
Translation practice and theory

Language teaching / learning

- ESL Teaching (English as Second Language)
- LSP Teaching (Language for Specific Purposes)

Type-Token Analysis

Each distinct word form is a type and each occurrence of a type counts as a token.



The Telugu corpus developed at the Language Engineering Research Centre (LERC), Department of Computer and Information Sciences, University of Hyderabad, India, hereafter referred to as LERC-UoH corpus, adds up to nearly 39 Million words, perhaps one of the largest corpora for any Indian language today.

END