Overview

- Introduction to the nature of syntactic representations. (Rambow, 15 minutes)
- Introduction to the morphology, syntax, and lexical semantics of Hindi and Urdu. (Sharma, 40 minutes)
- The morphological representation for Hindi and Urdu, including encoding issues, tokenization, part-of-speech tags, and morphological representation. (Sharma and Rambow, 20 minutes)
- The dependency representation (DS) for Hindi and Urdu syntax: principles, representation, and examples. (Sharma, 25 minutes)
- The lexical semantic representation (PB) for Hindi and Urdu: principles, representation, and examples. (Vaidya, 25 minutes)
- The phrase structure representation (PS) for Hindi and Urdu syntax: principles, representation, and examples. (Rambow, 25 minutes)
- Sample initial experiments in Hindi and Urdu NLP using the HUTB. (Sharma and Rambow, 15 minutes).

Representing Tokens, Morph Analysis, POS and Chunks in The Hindi/Urdu Treebanks

Outline

- Tokenization
- Morphological Representation
- POS tagging
- Chunking
- Inter-chunk dependency annotation
- Intra-chunk dependencies

Tokenization

- Automatic
- Issues
  - Compounds
  - Punctuations

For example,

usa	*ladake*	*ne*	*kelaa*	*khaayaa*	*thaa*
that	boy	erg	banana	eat-perf	past

Tokenization

Represented in SSF

ADDR	TOKEN
1	usa
2	laDake
3	ne
4	kelA
5	khAyA
6	thA
7	.

Tokenization: Issues

- Punctuations

All punctuations to be tokenized

- Compounds

BAI-bahana (brother-sister), bAlikA-vixyAlaya (girl-school)

- Compounds internally contain a punctuation
- Are productive
- Morphological analysis of the members of the compounds
- The issue, whether to create a single token
- Decision
- Create three tokens
- Mark the hyphen as 'JOIN'

Morph Analysis and its Representation

'af' defines the *composite attribute consisting of root, category, gender, number, person, case, tam (tense,aspect,modality)/vibhakti(case marker), suffix*

ADDR_	TKN_	OTHR
1	usa	<fs af='vaha,pr,,,'>
2	laDake	<fs af='laDakaa,n,m,sg,3,o,,'>
3	ne	<fs af='ne,psp,,,,,,'>
4	kelaa	<fs af='kelaa,n,m,sg,3,o,,,,'>
5	khaayaa	<fs af='khaa,v,m,sg,any,,yaa,'>
6	thaa	<fs af='kelaa,v,e,,,'>
7	.	<fs as='&STOP,punc,,,,,'>

POS Tagging

- ILMT POS Tagsets adopted
- Total 26 tags

ADDR	TKN	CAT	OTHR
1	usa	PRP	<fs af='vaha,pron…'>
2	laDake	NN	<fs af='laDakA,noun…'>
3	ne	PSP	<fs af='ne,psp…'>
4	kelA	NN	<fs af='kelA,noun…'>
5	khAyA	VM	<fs af='KA,verb…'>
6	thA	VAUX	<fs af='kelA,verb…'>
7	.	SYM	<fs as='&STOP,punc,,,,,'>

Chunking

- Chunking is introduced to save the effort in manual tagging
- Dependency relations are marked between the chunk heads
- Chunking restructures the tree, i.e.,

ADDR_	TKN_	CAT_	OTHR
1	((	NP	
1.1	usa	PRP	<fs af='vaha,pron…'>
1.2	laDake	NN	<fs af='laDakA,noun…'>
1.3	ne	PSP	<fs af='ne,psp…
	))		
2	((	NP	
2.1	kelA	NN	<fs af='kelA,noun…'>
	))		
3	((	VG	
3.1	khAyA	VM	<fs af='KA,verb…'>
3.2	thA	VAUX	<fs af='kelA,verb…'>
4.	((	BLK	
4.1	.	SYM	<fs as='&STOP,punc,,,,,'>
	))		