Report on Dependency Parsing in Marathi

- 2018114005, 2018114015

Status of Dependency Parsing in Marathi

Dependency Parsing in Marathi or any other Indian Language is really tricky because of major reasons like Projectivity and Morphological features (Marathi and Hindi have messy gender systems, and the SPMRL paper notes that it makes dependency parsing difficult).

Currently a paper notes the for Arc Standard values for Dependency Parsing in Marathi as 66.02 (Transition-based Parsing with Lighter Feed-Forward Networks). Another notes these values -

	Marathi											
f1	34.81	59.92	39.29	34.06	59.11	38.5	34.83	60.24	39.1	33.45	58.63	38.24
f2	64.25	83.52	68.79	62.57	81.15	67.38	63.96	83.38	68.44	61.98	80.74	66.94
f3	66.27	84.6	69.67	65.22	83.66	69.2	66.08	84.58	69.45	65.11	83.41	69.14
f4	70.33	86.99	74.1	68.12	84.33	72.69	70.39	87.04	74.04	68.07	84.48	72.61
f5	70.47	87.32	74.26	68.42	85.18	72.44	70.25	87.19	74.15	68.0	84.92	72.07
f6	71.01	87.72	74.75	69.56	86.28	73.42	70.46	87.5	74.18	68.45	86.06	72.3
f7	**71.56**	**88.05**	**74.95**	**69.75**	**86.41**	**73.52**	**70.97**	**87.69**	**74.47**	**69.01**	**85.98**	**72.82**

Table 4: Parsing accuracies of our neural network based parser for all 5 languages. Auto development and test set contain predicted POS and chunk tags. Gloss of the features are f1 = POS only, f2 = f1+ suffix, f3 = POS + word, f4 = f3 + suffix , f5 = f4+ PSP, f6 = f5 + chunk, f7 = f6 + GNP

We can see how challenging of a task this is.

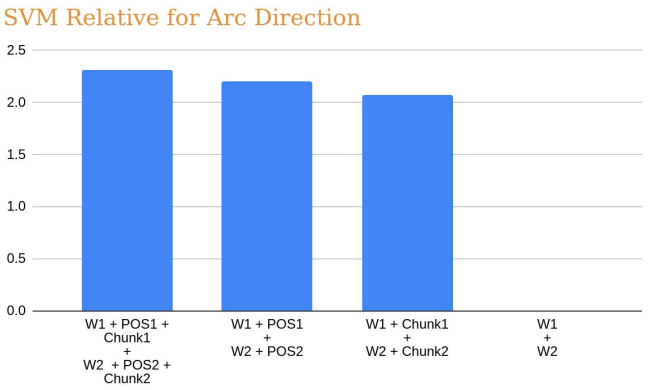
Furthermore, there has been no public implementation of the Arc Eager Parsing for Marathi that we could compare with (at least to our knowledge), thus we resorted to find out what pre conditions could be applied to the standard Arc Eager implementation by looking at Hindi parallels, as described below.

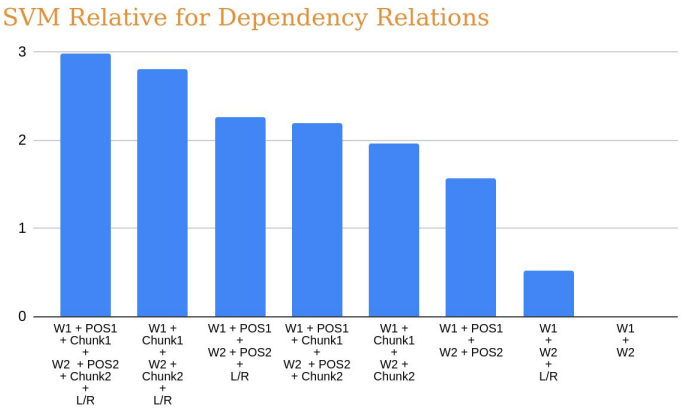
Arc Eager

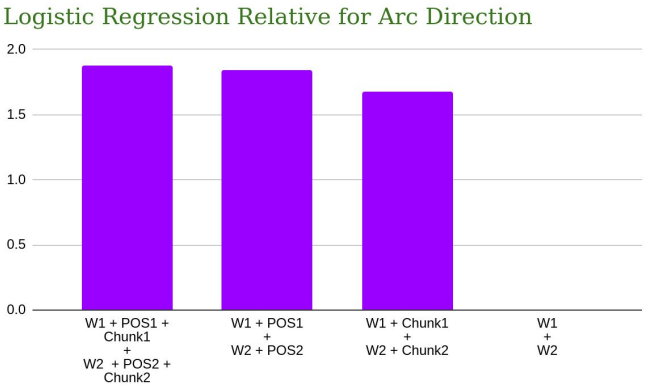
The implementation was mostly based on the standard algorithm in Jurafsky. The change was only in the REDUCE operation. Initially check if a Right Arc or a Left Arc is possible. If No then check for REDUCE operation. For REDUCE operation, first check if the top of the buffer has L/R relation with any of the elements in the stack except the top. If yes, then check if the top of stack is already a head of some other element. If the answer is yes in both the cases, then apply REDUCE, else apply SHIFT.

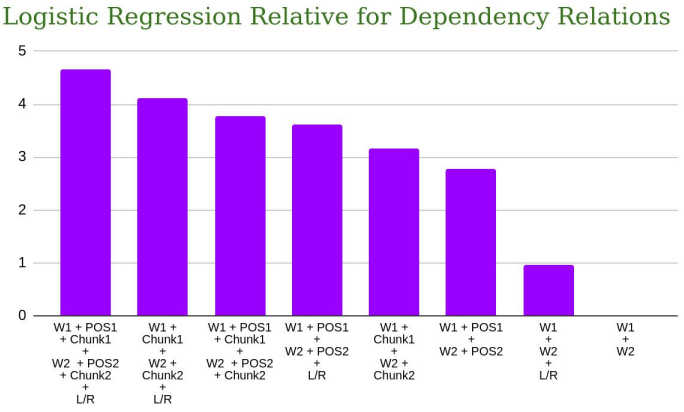
The sentences which were found out to be non parsable were those sentences which had projectivity issues. If all of the arcs can be drawn without any intersection of any of the arcs (in one plane), then the sentence was parsable. Else, if at least two of the arcs intersected, then the sentence was non-parsable and Arc Eager removed the sentence from the corpus.

Findings and Comparisons





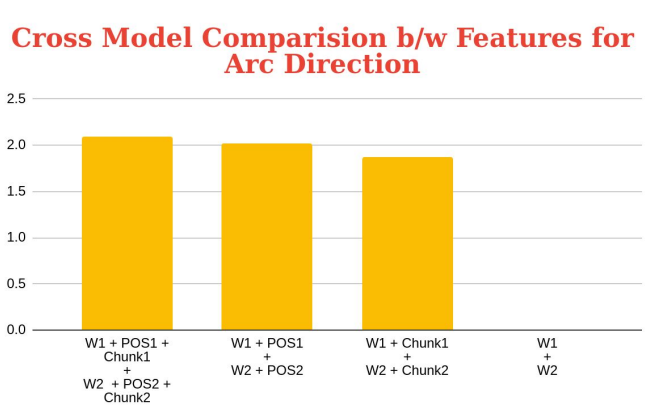


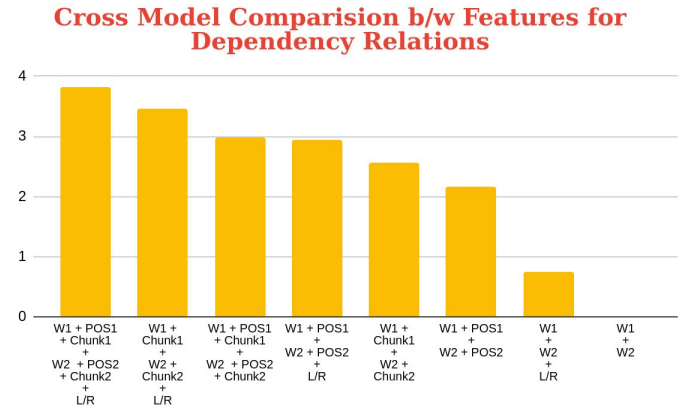


From the above comparisons we can easily see that -

- POS Tag outperforms Chunk Tag as a feature on determining Arc Direction
- Chunk Tag outperforms POS Tag as a feature on determining Dependency Relations

This is easily verifiable by the fact that a Chunk in general provides more ***Semantic*** information than a POS Tag and thus helping identify Dependency Relations better.
This is reinforced by our later inclusion of some more information like Gender, Number, etc. which in Indian Languages causes redundancy that helps identify dependency relations better.





All in all, independent of the Oracle ML model used -

- Arc Direction prediction has little to no use for Chunk Tags
- Dependency Relations predictions using the Arc Direction as a feature outperform their counterparts without Arc Direction input as a feature, though Arc Relations alone aren't useful at all.

Dependency Parsing has been following the *"first determine direction and then label them"* technique for quite some time now, this shows clearly in our findings too.

While that is true, we deem that Arc Direction prediction in Marathi should be done using POS Tag and the Word Form itself.
We can see that adding more information like Gender, Number, etc. to POS Tag is causing predictions to go down in accuracy, while Chunk Tags don't harm, but they don't help either.

In conclusion, our best features for finding Dependency Relations were -

Chunk Tag,Word form itself, Morphological Features, and Arc Directions (76.92 % from SVM Oracle)

And for Arc Directions were -

Word form itself + POS Tags + Chunk Tags *(which was nearly equal to* ***Word form itself + POS Tags)***

And an SVM Oracle performed better than Logistic Regression in most cases.

Below are the tables with the Model wise breakdown of values. The tables in the beginning include Morph features, while the latter tables are the models where we explored all possible combinations of features except Morphological features.
This was because we drew our hypothesis from the combinations except Morph features and then ran the best performing model combinations with Morph features to see if the above hypothesis about Drel being enriched with Semantic/ Redundancy features is better ? and whether It is POS Tag itself that is driving the Arc Directions, as both are very structural in nature.

Model. No	**No. of Features**	**Predicting?**	**Features**	**SVM**	**Logistic Regression**
13	8	L/R/U	W1 + POS1 + Chunk1 + Morph1 + W2 + POS2 + Chunk2 + Morph2	79.91	79.76
16	6	L/R/U	W1 + POS1 + Morph1 + W2 + POS2 + Morph2	79.61	79.72
Model. No	**No. of Features**	**Predicting?**	**Features**	**SVM**	**Logistic Regression**
14	8	drel	W1 + POS1 + Chunk1 + Morph1 + W2 + POS2 + Chunk2 + Morph2	76.25	Not calculated

15	9	drel	W1 + POS1 + Chunk1 + Morph1 + W2 + POS2 + Chunk2 + Morph2 + L/R	76.89	Not calculated
17	7	drel	W1 + Chunk1 + Morph1 + W2 + Chunk2 + Morph2 + L/R	76.92	Not calculated

Arc Direction

Model. No	No. of Features	Features	Predicting?	SVM	Logistic Regression	SVM relative	Logistic Regression Relative	Average over Models
1	6	W1 + POS1 + Chunk1 + W2 + POS2 + Chunk2	L/R/U	80.26	80.27	2.31	1.88	2.095
7	4	W1 + POS1 + W2 + POS2	L/R/U	80.16	80.23	2.21	1.84	2.025
10	4	W1 + Chunk1 + W2 + Chunk2	L/R/U	80.02	80.07	2.07	1.68	1.875
4	2	W1 + W2	L/R/U	77.95	78.39	0	0	0

DREL

Model. No	No. of Features	Features	Predicting?	SVM	Logistic Regression	SVM relative	Logistic Regression Relative	Average over Models
3	7	W1 + POS1 + Chunk1 + W2 + POS2 + Chunk2 + L/R	drel	76.51	73.05	2.99	4.65	3.82

12	5	W1 + Chunk1 + W2 + Chunk2 + L/R	drel	76.33	72.52	2.81	4.12	3.465
2	6	W1 + POS1 + Chunk1 + W2 + POS2 + Chunk2	drel	75.71	72.17	2.19	3.77	2.98
9	5	W1 + POS1 + W2 + POS2 + L/R	drel	75.78	72.01	2.26	3.61	2.935
11	4	W1 + Chunk1 + W2 + Chunk2	drel	75.48	71.56	1.96	3.16	2.56
8	4	W1 + POS1 + W2 + POS2	drel	75.09	71.17	1.57	2.77	2.17
6	3	W1 + W2 + L/R	drel	74.04	69.36	0.52	0.96	0.74
5	2	W1 + W2	drel	73.52	68.4	0	0	0

References :

The following papers, tutorials, algorithms etc.

- Statistical Parsing of Morphologically Rich Languages (SPMRL) What, How and Whither
- Transition-based Parsing with Lighter Feed-Forward Networks
- Unity in Diversity: A unified parsing strategy for major Indian languages

- *NLP Programming Tutorial 12 -Dependency Parsing, Graham Neubig*
- *Transition-based dependency parsing, Sara Stymne*
- *Statistical dependency analysis with support vector machines, Yuji Matsumoto ;*
- *Universal Dependency Parsing from Scratch, Manning ;*
- *Transition-based Dependency Parsing with Rich Non-local Features, Nivre*
- *MaltParser: A Data-Driven Parser-Generator for Dependency Parsing, Nivre and Hall*
- *Algorithms for Deterministic Incremental Dependency Parsing, Nivre*
- *Stanford typed dependencies manual, Manning ; AnnCorra, LTRC*