UNSUPERVISED CHUNKING ALGORITHMS FOR HINDI: COURSE PROJECT

LANGUAGE TYPOLOGY AND UNIVERSALS

PRAJNEYA KUMAR
JAYANT PANWAR

OUTLINE OF THE PRESENTATION

- ★ Chunking
- ★ Literature Survey
- ★ Base Model Recap
- ★ Base++ Model
- ★ Evaluations and Results
- ★ Conclusion and Improvements

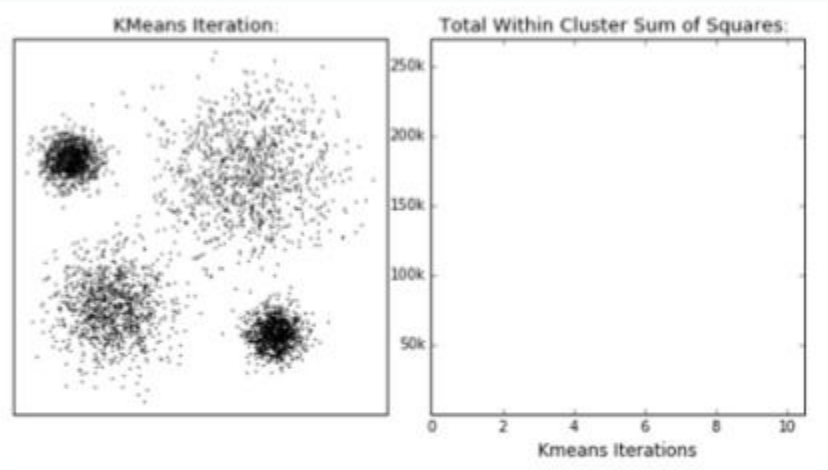
CHUNKING

- ➔ Process of identifying **chunks,** which are non-overlapping and non-recursive regions of text which contain a head and related function words and modifiers
- ➔ Has a wide range of use in larger NLP tasks such as Information Extraction, Named Entity Recognizers, Question-Answering, etc.
- ➔ Hindi does not have any major chunkers/shallow-parsers. Dependency parsers available though
- ➔ Supervised algorithm need input-output pairs. Not good for extended period of time as languages evolve. Unsupervised algorithm best way out of this dilemma.

LITERATURE SURVEY

- Shallow Parsing Pipeline for Hindi-English Code-Mixed Social Media Text; https://arxiv.org/pdf/1604.03136.pdf
- Chunking in NLP Decoded; https://towardsdatascience.com/chunking-in-nlp-decoded-b4a71b2b4e24
- K-Means Clustering; https://towardsdatascience.com/k-means-clustering-8e1e64c1561c
- Unsupervised Chunking Based on Graph Propagation from Bilingual Corpus; https://downloads.hindawi.com/journals/tswj/2014/401943.pdf
- Hierarchical Clustering with Python and Scikit-Learn; https://bit.ly/3e7EGFW

BASE: K-MEANS CLUSTERING



Chunking, or to say, Local Word Groupings, on their very first intuition, are closely related to the distance between words.

K-means Clustering is an unsupervised learning algorithm of vector quantization that aims to partition n samples into k segregated clusters.

BASE: K-MEANS CLUSTERING

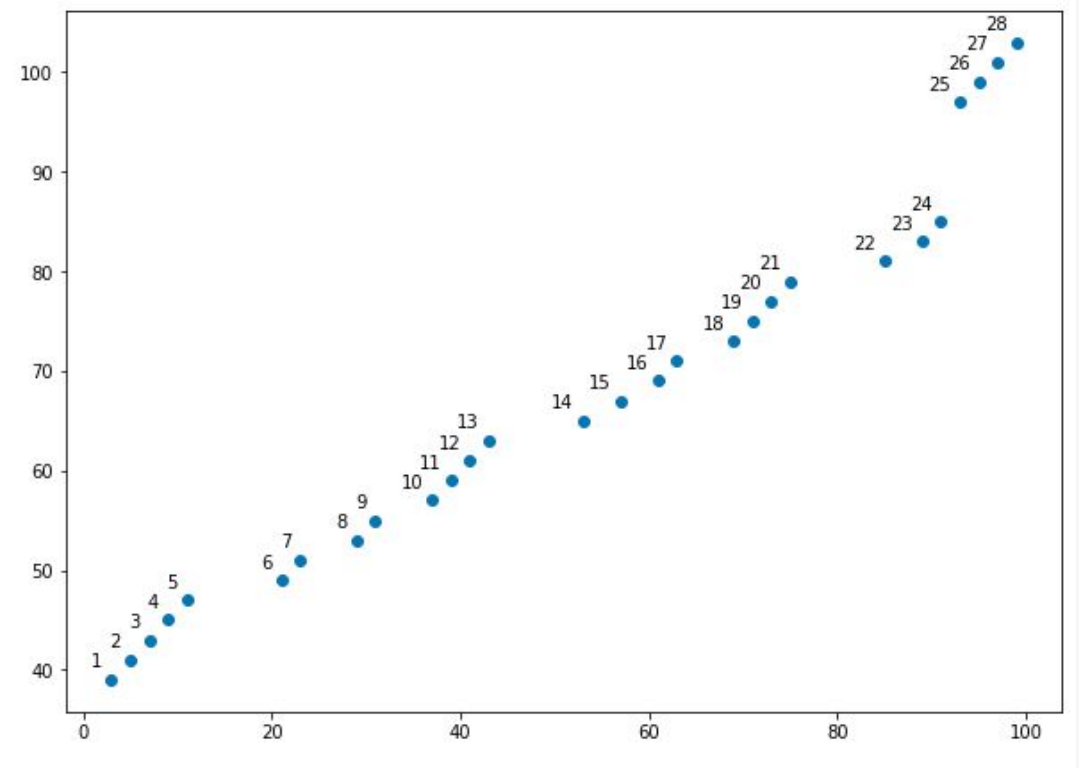
1. Assume number of centroids
2. Randomly allocate word as centroids
3. Allocate word groupings according to distance from head
4. Repeatedly change the head centroid to the first noun/verb that is found in each cluster
5. Report final clusters

BASE++: HIERARCHICAL CLUSTERING

- ★ Unsupervised algorithm used to cluster unlabeled data
- ★ Groups data points with similar characteristics
- ★ Two types: Agglomerative (bottom-up) and Divisive (top-down)
- ★ Utilized Agglomerative as it is same as hands on paper approach which is done manually

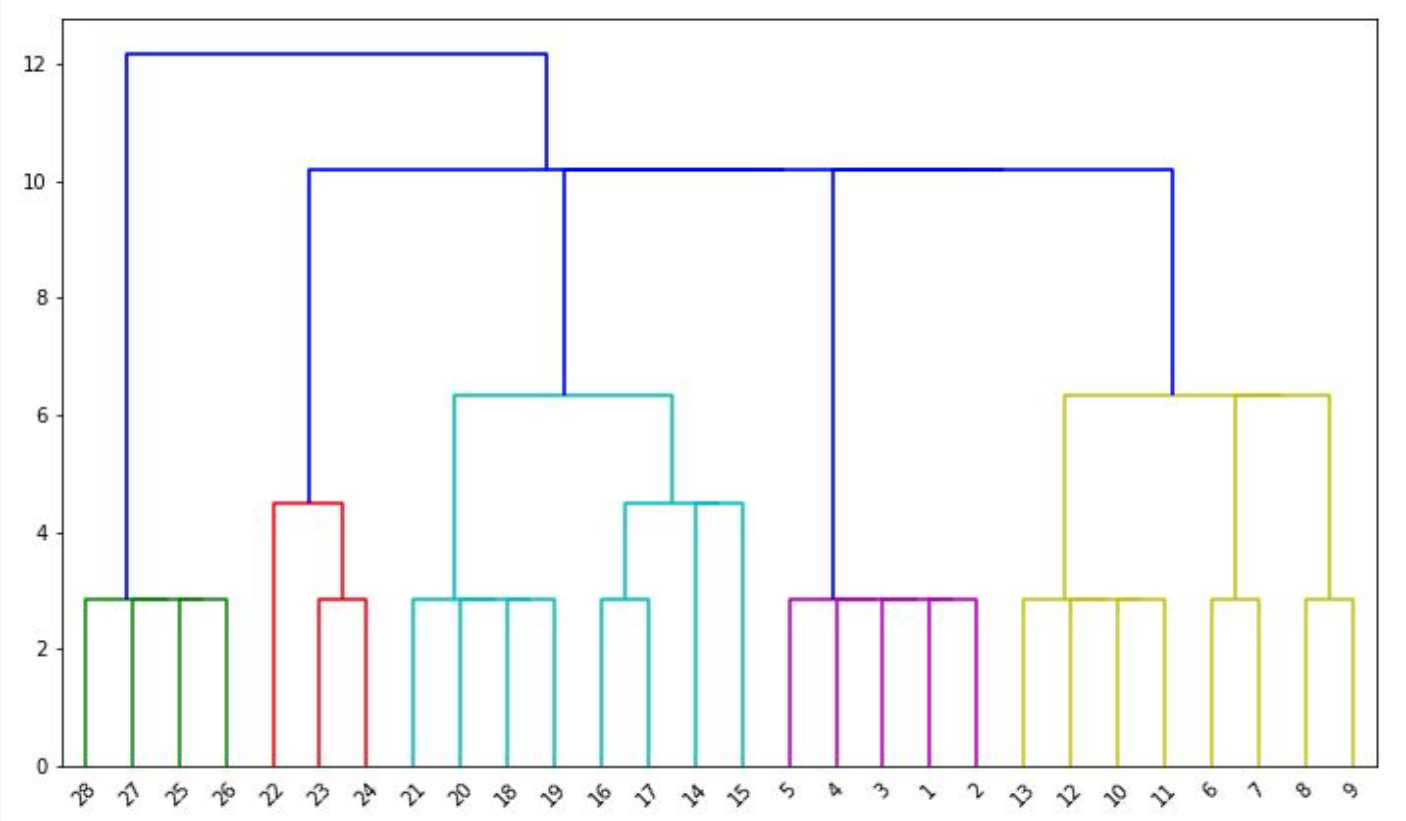
BASE++: HIERARCHICAL CLUSTERING

- ★ Firstly, convert the words into (x,y) coordinates of a graph.
- ★ x: index of the word + index of the closest noun
- ★ y: index of the word + index of the closest verb
- ★ Euclidean distancing used for determining similarity



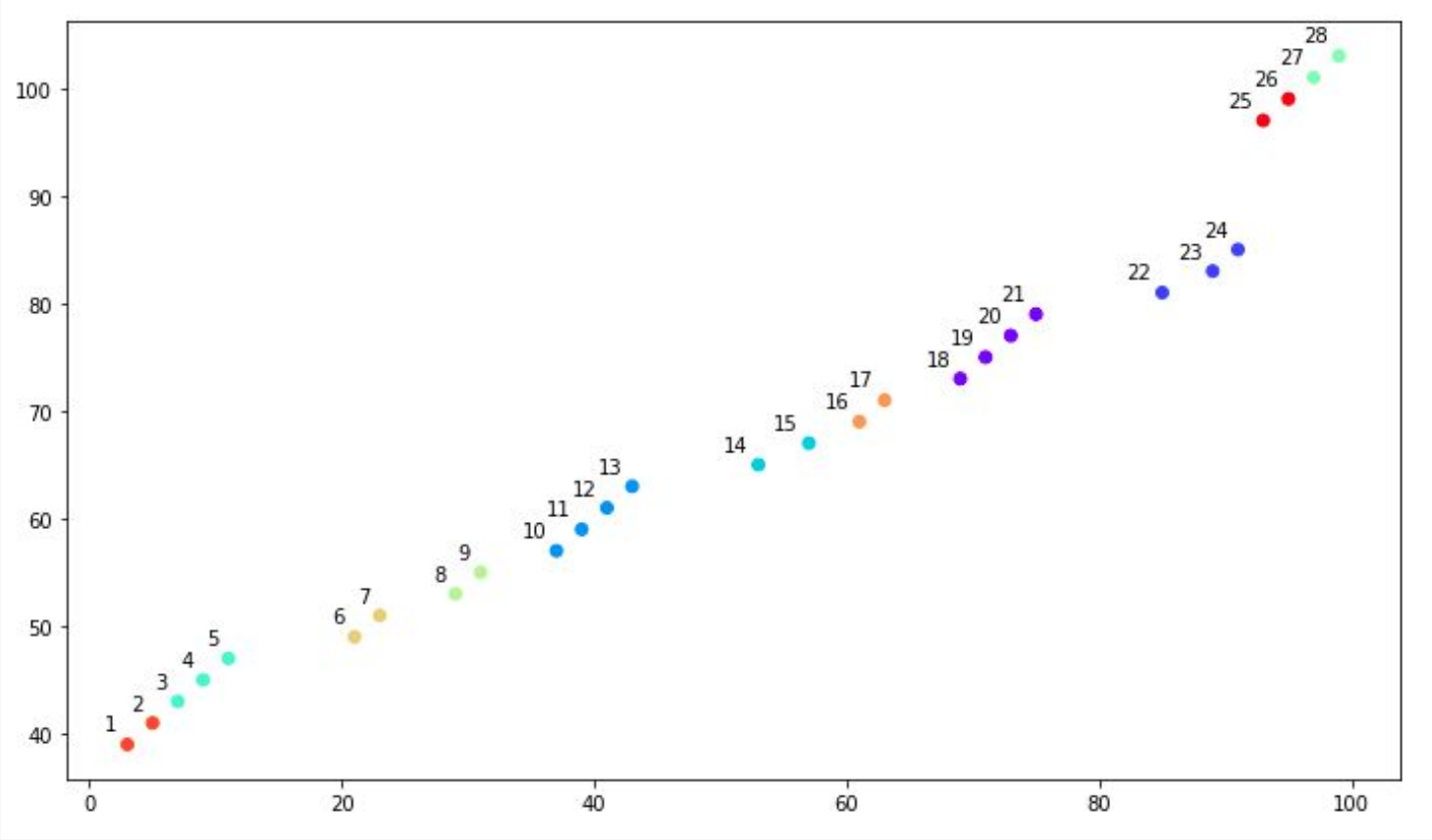
BASE++: HIERARCHICAL CLUSTERING

- ★ Convert the data points into dendrogram
- ★ Look for the longest vertical line
- ★ Draw a horizontal line at the base of the longest vertical line. The number of vertical lines it crosses will give us the number of clusters that should be found.



BASE++: HIERARCHICAL CLUSTERING

- ★ Create an instance of AgglomerativeClustering class of sklearn.cluster package
- ★ Use ‘ward’ linkage
- ★ Give the number of clusters same as determined from dendrogram



EVALUATION

Initially we had planned to use NLTK.chunk() library to chunk our data and create a gold standard for our model's evaluation.

Due to the False Positives problem, we obtained a manually annotated LWG dataset for our sentences, and used it as the Gold standard.

EVALUATION: BASE

- The average overall accuracy of our model is **71.04%**
- The average accuracy of Noun Phrase allocation for our model is **78.25%**

- Random allocation of head centroid changes results
- Assumption of number of centroid leads us to inaccuracies
- Only NPs and VPs can be looked at
- Simultaneous occurrence of two consecutive nouns can be a problem
- Equidistant words have to be differentiated by POS Tags

इसके NP	इसके NP	इसके B-NP
चारों NP	चारों NP	चारों B-NP
ओर NP	ओर OTHER	ओर I-NP
दीवार NP	दीवार VP	दीवार B-NP
है NP	है VP	है B-VGF
और OTHER	और VP	और B-CCP
बीच OTHER	बीच VP	बीच B-NP
में OTHER	में VP	में I-NP
एक NP	एक NP	एक B-NP
तालाब NP	तालाब NP	तालाब I-NP
है NP	है VP	है B-VGF
। BLK	। BLK	। B-BLK

EVALUATIONS: BASE

- The average overall accuracy of our model is **71.04%**
- The average accuracy of Noun Phrase allocation for our model is **78.25%**

- Random allocation of head centroid changes results
- Assumption of number of centroid leads us to inaccuracies
- Only NPs and VPs can be looked at
- Simultaneous occurrence of two consecutive nouns can be a problem
- Equidistant words have to be differentiated by POS Tags

इसका B-NP	इसका B-NP
प्रवेश B-NP	प्रवेश B-NP
द्वार B-NP	द्वार I-NP
दो OTHER	दो B-JJP
मंजिला OTHER	मंजिला I-JJP
है B-VP	है B-VGF
। BLK	। B-BLK

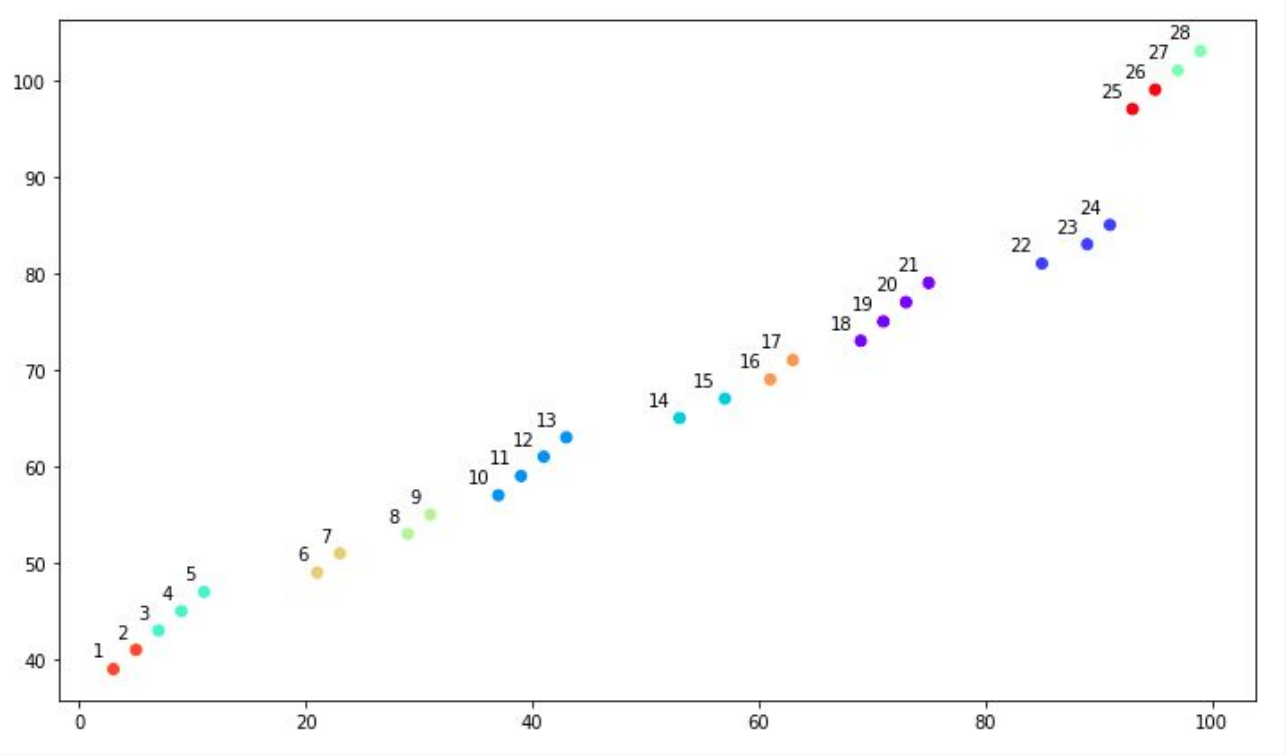
EVALUATIONS: BASE++

- ★ Since finding the number of clusters is a manual task, chunking for all the sentences could not be done
- ★ 20 sentences were evaluated manually
- ★ 4 matched with the gold standard. Rest had either the number of chunks wrong or the words in the chunk wrong
- ★ Surprisingly, the algorithm managed to find adjectival and prepositional phrases at times even though only trained for noun and verb phrases

```
1   शौकत      N_NNP
2   महल N_NNP
3   के   PSP
4   सामने    N_NST
5   बड़ी JJ
6   झील N_NN
7   के   PSP
8   किनारे   N_NN
9   स्थित    JJ
10  वास्तुकला    N_NN
11  का   PSP
12  यह   DM_DMD
13  खूबसूरत JJ
14  नमूना    N_NN
15  कुदसिया N_NNP
16  बेगम     N_NNP
17  के   PSP
18  काल N_NN
19  का   PSP
20  है   V_VM
21  जिन्हें PR_PRL
22  गोहर     N_NNP
23  बेगम     N_NNP
24  भी   RP_RPD
25  कहा V_VM
26  जाता     V_VAUX
27  था   V_VAUX
28  ।    RD_PUNC
```

EVALUATIONS: BASE++

- ★ gold standard: 10 NPs, 1 JJP, 2 VGF
- ★ Base++ model: 6 NPs, 1 JJP, 3 VGF : 1 NP 1VP matched




```
1   शौकत      N_NNP
2   महल N_NNP
3   के  PSP
4   सामने     N_NST
5   बड़ी JJ
6   झील N_NN
7   के  PSP
8   किनारे   N_NN
9   स्थित     JJ
10  वास्तुकला     N_NN
11  का  PSP
12  यह  DM_DMD
13  खूबसूरत JJ
14  नमूना     N_NN
15  कुदसिया N_NNP
16  बेगम      N_NNP
17  के  PSP
18  काल N_NN
19  का  PSP
20  है  V_VM
21  जिन्हें PR_PRL
22  गोहर      N_NNP
23  बेगम      N_NNP
24  भी  RP_RPD
25  कहा V_VM
26  जाता      V_VAUX
27  था  V_VAUX
28  ।   RD_PUNC
```

BASE VS BASE++

- ➔ The Base model clearly outperformed our Base++ model
- ➔ Base model's efficiency at detecting the NPs was clearly better. The overall accuracy of the Base model was greater as well
- ➔ Some reasons as to why Base model is performing better:
  - ❑ Determining the number of cluster by using Linguistic knowledge
  - ❑ Utilizing linguistic knowledge to determine the components of chunks and chunk boundaries
  - ❑ More possibilities should be considered rather than just noun distance and verb distance for distance matrix in Base++ model

IMPROVEMENTS

- A better algorithm to approximate K in Base model (K-Means Clustering algorithm) will increase the accuracies as well

- Similarly, automated methods like Elbow method and Inconsistency method can be tried in Base++ model. Manual selection with linguistic knowledge would also go a long way

- More linguistic knowledge can be applied in Base++ model. As of now only verb phrases and noun phrases being considered

CONCLUSION

- We were able to not only build introductory level chunkers for Hindi using Unsupervised learning algorithms but we were also able to detect fallacies and suggest improvements in our models

- If parallel tagged corpora were available, Graph propagation algorithm could also have been used to develop an unsupervised Chunker similar to the one developed by Ling Zhu et al.

Link to Google Slide:
https://docs.google.com/presentation/d/10vl5IkEftb2SSH9YArlw7qpi_8yZu0WivbcLtO_m6OE/edit?usp=sharing