Automatic extraction of Selectional Constraints for verb frames in Hindi

-Abhilash I , G.V.Sivakumar
IIIT Hyderabad

Introduction

- What are Selectional constraints ??

  Selectional constraints specify the semantic classes acceptable in syntactic structures.

  Eg: <Verb: Eat> <subj: Animate> <obj: Edible>

Motivation

- Hindi being a free word order language, selectional constraints help in disambiguating the dependency parse.

  Eg: rAm pHala khatA heM

  (ram) (fruit) (eat) (present_tense)

  pHala rAm khatA heM

  (fruit) (ram) (eat) (present_tense)

- Selectional constraints can validate the documents semantically.

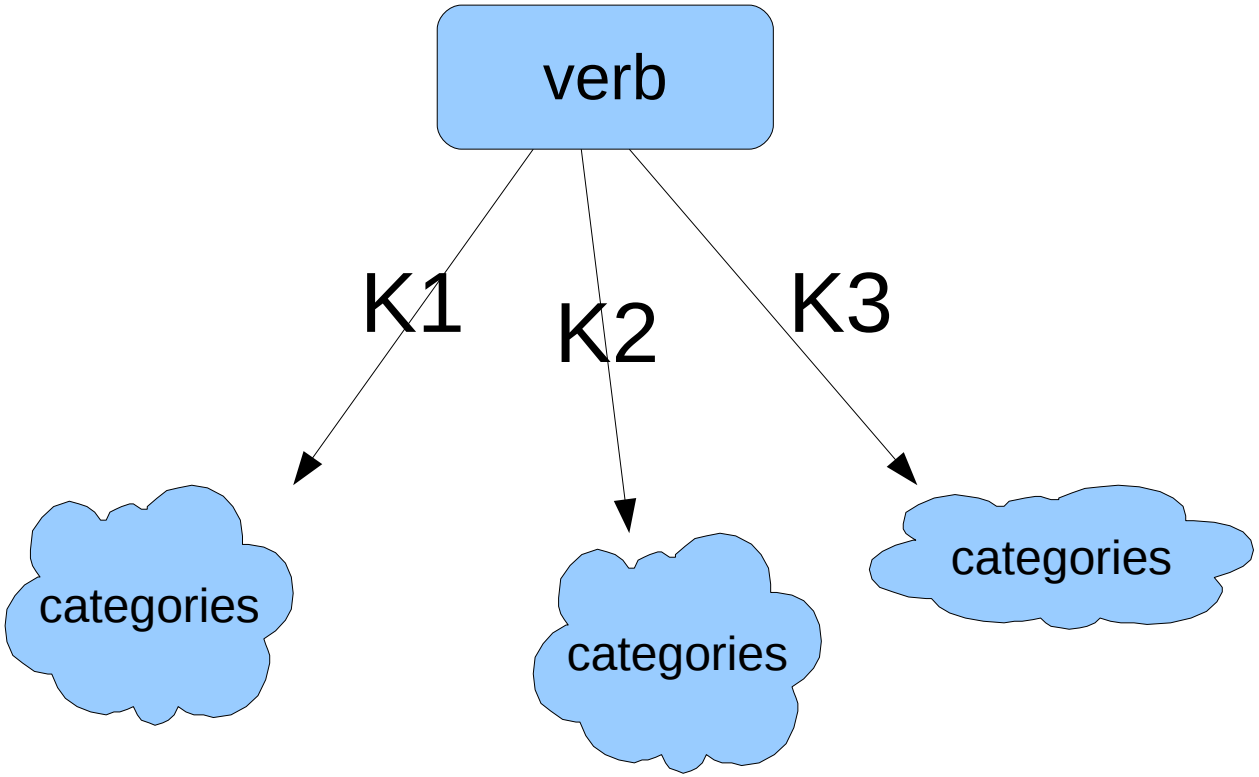
Related Work

- Smoothing of automatically generated selectional constraints (Ralph Grishman, John Sterling 1993)
- A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology Acquisition (David Faure & Claire Nedellec)

Resources

- Dependency Tree Bank
  - 1393 sentences
  - 296 unique verbs
- Hindi Wordnet
  - To extract Ontological Categories of a lexical item

Goal



Example

For

Verb: **KA(eat)** and **karaka K2 (patient)**

Selectional constraints =

{ Edible (Higher Probability) , Animate }

First approach #alg1

- Verb **KA(eat)** and **karaka k2**
  - **Ama(mango) , cat{Ama}= {Fruit ,Tree, Edible }**
  - **miTAI(Sweet), cat{mitAI)= {Edible, Artifact}**
  - **The selectional constraints of eat are**
    - **cat{Ama} ∩ cat{mitAI} i.e { Edible }**

Pitfalls of this approach

- This algorithm assumes that if a category have to be selected it should be present in atleast two lexical items. This will fail if it is not so.
- Eg : For verb **“pI”(drink)**

  lex1 **pAni(water)**

  cat[water]= { xrava(liquid) }

  lex2 **soda**

  cat[soda]= { drink }

  *Here the final output is NULL. But we know drink is a **subtype** of liquid*

Another approach #alg2

- **Ama(mango) , cat{Ama}= {Fruit ,Tree, Edible }**
  - P[Fruit/mango]=1/3, P[Tree/mango]=1/3, p[Edible/mango]=1/3
- **miTAI(sweet), cat{mitAI)= {Edible, Artifact}**
  - P[Edible/sweet]= 1/2, P[Artifact/sweet]= ½
- **Therefore, the selectional constraints of eat are**
  - **score[Fruit]= 1/3, score[Edible]= 1/3+ 1/2, score[Tree]= 1/3**

- Better results were found compared to #alg1
- This algorithm was able to resolve some ambiguities in parse
- Sample output for verb pI(drink) & karaka k2(patient)
  - ('xrava ', 'Liquid') 50.0
  - ('prAkqwika vaswu ', 'Natural Object') 33.3333333333
  - ('vaswu ', 'Object') 16.6666666667
- **Pitfalls of this approach**
  - If the wordnet gives many classes, the score is distributed over all this classes.

A pitfall for Verb--> KA (eat) and karaka k2(patient) the selectional constraints are

[rotI, capAwI] #2567 6.66666666667

[miTAI, mITA, miRtAnna] #7647 6.66666666667

[pakavAna, pakvAna, pakkA_KAnA, pakkI_rasoI, pakkI-rasoI, vyaMjana, byaMjana] #6594 6.66666666667

[Soka, gama, gZama, gamI, gZamI, raMja, avasAxa, soga, aMxoha, anxoha, aBiRaMga, aBiRafga] #5293 6.66666666667

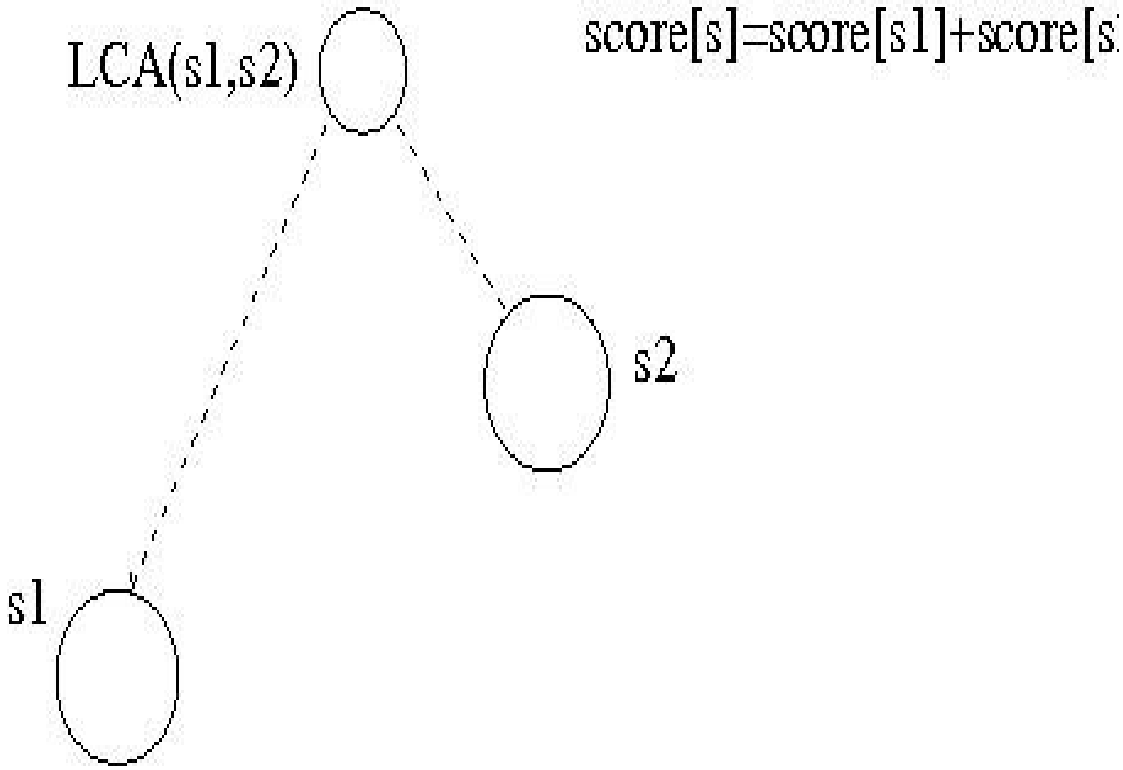
[biskuta, biskita] #2077 3.33333333333

[pUrI, pUdZI] #428 2.22222222222

Better approach

- Finding the distance between classes formed from #alg 2 so that similar classes combine to give a parent class.

Intuition



Algorithm 3 Refining using the scores obtained from previous algorithm

1: $C = Cat[v_{K_i}]$
2: $changes = true$
3: **while** $changes$ is $true$ **do**
4: $changes = false$
5: $< c_i, c_j, minDist >= \underset{c_i, c_j \in C \& c_i \neq c_j}{\text{argMin}} \; dist(c_i, c_j)$
6: **if** $minDist \leq Threshold$ **then**
7: $changes = true$
8: $C = \{C - \{c_i, c_j\}\} \cup LCA(c_i, c_j)$
9: $score[LCA(c_i, c_j)] += score[c_i] + score[c_j]$
10: **end if**
11: **end while**

- Verb *KA* 'eat' , k2(patient) and **threshold=1**
  - [rotI, capAwI] #2567 6.66666666667
  - [miTAI, mITA, miRtAnna] #7647 6.66666666667
  - [pakavAna, pakvAna, pakkA_KAnA, pakkI_rasoI, pakkI-rasoI, vyaMjana, byaMjana] #6594 6.66666666667
  - [Soka, gama, gZama, gamI, gZamI, raMja, avasAxa, soga, aMxoha, anxoha, aBiRaMga, aBiRafga] #5293 6.66666666667
  - [biskuta, biskita] #2077 3.33333333333
  - [pUrI, pUdZI] #428 2.22222222222

- Verb KA(eat) , karaka k2(patient) and **threshold=2**

[KAxya_vaswu, KAxya_paxArWa, KAxyavaswu, KAxyapaxArWa, AhAra, KAxya, Bojya_paxArWa, AhAra_paxArWa, anna] #20	29.4444444444
[vyakwi, mAnasa, SaKZsa, SaKsa, jana, baMxA, banxA] #196	8.88888888889
[vaswu, cIjZa, cIja] #923	6.66666666667
[vacana, vANI, bolI, bANI, bAnI] #2934	6.66666666667
[Soka, gama, gZama, gamI, gZamI, raMja, avasAxa, soga, aMxoha, anxoha, aBiRaMga, aBiRafga] #5293	6.66666666667
[vaswu-BAga, vaswu-aMga, vaswu_BAga, vaswu_aMga] #3136	5.0
[gAya, gaU, gEyA, go, gO, Xenu, suraBi, rohiNI] #4441	3.3333333333
[biskuta, biskita] #2077	3.3333333333

- For verb KA(eat) , karaka k2(patient) and **threshold=3**

[vaswu, cIjZa, cIja] #923	38.3333333333
[vyakwi, mAnasa, SaKZsa, SaKsa, jana, baMxA, banxA] #196	12.2222222222
[vacana, vANI, bolI, bANI, bAnI] #2934	6.66666666667
[Soka, gama, gZama, gamI, gZamI, raMja, avasAxa, soga, aMxoha, anxoha, aBiRaMga, aBiRafga] #5293	6.66666666667
[vaswu-BAga, vaswu-aMga, vaswu_BAga, vaswu_aMga] #3136	5.0
[biskuta, biskita] #2077	3.33333333333

Observations

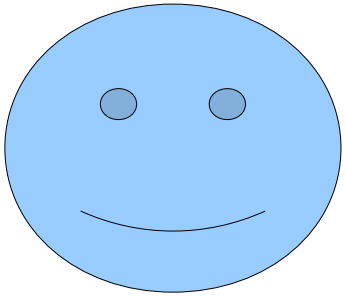
- The results for low threshold values are too specific
- The results for high threshold values are too generic
- Medium threshold values gave good results.

Future work

- Using cross POS links in wordnet, classes can be filtered so that some classes can be discarded in a particular context
  - Eg: categories(bat)= [mammal, playing instrument]
  - In the context of play, mammal can be discarded
- Threshold function
- Evaluation of this approach

Acknowledgments

Thanks to Prof Rajeev Sangal, Dr Dipti Misra Sharma, Dr Srinivas Bangalore, Samar, Rafiya and winterschool administration.



Thank you