PROJECT

POS TAGGING
USING RESOURCE RICH LANGUAGE

Suman Dowlagar(201307678)

&

Nirmal Surange(201499520)

AIM

Using POS tagged resource rich language tag the resource poor language

(example resource rich data taken : Hindi)

(example resource poor data taken : Marathi)

Assumption:

Languages are fairly similar

Paper Followed

Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections

By

Dipanjan Das & Slav Petrov

Resources Used

- Parallel Hindi and Marathi data consisting of 50,000 sentences taken from Health and tourism domains
- Hindi data is POS tagged
- Giza++ version 1.0.7

Programming languages used & Model Tested on:

Programming languages used :

- Python
- java

Model tested on:

- Ubuntu 14.04

Approach Overview:

- We use graph-based approach for clustering the similar Marathi words and use Alignment tools to transfer tags of Hindi corpus to Marathi corpus

Algorithm 1 (Calculation of trigram similarity

- First we calculate the bilingual similarity graph.
- Graph labeling :
  - we use two types of vertices in our graph:
  - Marathi trigram vertices(using individual words as the vertices throws away the context necessary to disambiguate),
  - Hindi unigram vertices.
- Graph construction(calculation of two similarity functions.) :
  1. Using features of the trigrams, the edge weight between each two trigrams are calculated.
  2. Using unsupervised word alignment each Marathi trigram center word is aligned to hindi word.

Implementation procedure:

- Tokenization :
  - Preprocess the given hindi and marathi data(ie. Tokenizing and checking for errors)
- Extraction of all the uni,bi,tri and 5-grams : Why?
  - As tagging each word, has ambiguity, so left and right context is considered(so trigrams are taken out)
  - Bi-grams and 5-grams are used for calculating feature values for the trigrams.
  - These feature values are in the form of PMI scores.
  - (PMI says: How the words are related to each other)
  - Given by:

$$\mathrm{pmi}(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)}$$

Continued...

- Similarity Function :
  - Given the features of the trigrams, using those feature values find how similar the two trigrams are.
- Alignment :
  - For the given Hindi tagged corpus and Marathi untagged corpus we run the giza++ tool on both sides and calculate the cross alignments which represent high confidence alignments.

Continued...

- Extraction of Tags from the word alignments :
  - As Giza++ gives the word alignments, syntactically, Marathi word will be of the same category of the given Hindi word
  - So reflect the tag of Hindi word to Marathi word.
- Neighbor Calculation :
  - As the similarity matrix says more about the center word.
  - Given the similarity matrix we extract the center word from trigram word
  - This is done because,(in the cross alignment matrix some words will be having no high confidence alignments, so they will have no tags) if a word has no tag aligned to it, it can take the tag from one of its neighbors. We use 5 neighbors in the alignments.

Continued...

- RI COUNT:
  - This is calculated as Count(word_tag)/Count(word), this says for a word occurred N number of times, how many times it has occurred with the given tag.
- QI COUNT:
  - This gives the distribution of word for all the tags.
  - This works on the principle of Estimation maximization algorithm, so that the best possible score is obtained for a tag given a word. Maximum 10 iterations are considered here.

POS Projection

- POS INDUCTION:
- Now we have the tagged Marathi data as result.
- Taking counts of QI for the particular word, For the given untagged Marathi data, tag each marathi word with the highest QI score for each word.

- RESULTS:

- For the given input of the form:
  - स्वच्छ उच्छ्वास आणि दात हे आपले व्यक्तिमत्व खुलवतील .
  - दातांमुळे आपला आत्मविश्वाससुद्धा वाढतो .

- The output is of the form:
  - स्वच्छ_JJ उच्छ्वास_N आणि_CC दात_N हे_DM आपले_PR व्यक्तिमत्व_N खुलवतील_V
  - दातांमुळे_N आपला_PR आत्मविश्वाससुद्धा_N वाढतो_V

Results:

- **The approach is applied for 100 Marathi test sentences,** we compared it with the manually tagged gold standard data, we got **accuracy as 70.6%** which is good in the field of unsupervised POS tagging.

THANK YOU