Nirmal Surange & Dowlagar Suman

# POS TAGGING USING RESOURCE RICH LANGUAGES

AIM:

- For inducing unsupervised part-of-speech taggers for languages that have no labeled training data, but have translated text in a resource-rich language.

INPUT:

- Hindi tourism and health domain tagged data (As Resource Rich Language)
- Marathi tourism and health domain untagged data (As Resource Poor Language)
- Tags considered(12 tags) which are the basic tags, for better reliability.

PAPER FOLLOWED:

- Unsupervised Parts of Speech tagging with bilingual graph based projections By Dipanjan Das and Slav Petrov.

TOOLS USED:

- GIZA++ version 1.0.7 for word alignment of Hindi and Marathi data.

PROGRAMMING LANGUAGES USED:

- Python
- Java

MODULE TESTED ON:

- Ubuntu 14.04

PREVIOUS WORKS(IN THE FIELD OF POS TAGGING):

1. Dictionary based POS tagger (which takes POS category for a particular word from dictionary and applies it to the word) cons:words have ambiguous tags(no context used)
2. Feature based POS tagger (take the bound morphemes from the words extract the tag for that morpheme ie., bigger has “er” as suffix so it takes adjective)
3. Cluster the give words using k-means, the cluster of words behave similarly and take a same tag. Take one word from the cluster, find its tag in the dictionary and apply that tag to all the other words. Pros: very robust, Cons: can't resolve ambiguity.
4. For the above method with probability (stochastic based POS tagger) might resolve ambiguity and apply the best tag for the given word.
5. Hence the tag probabilities are learnt from the hindi tagged data set and marathi words are clusters and hindi tags applied to the Marathi dataset. With this we have one tag for one word with good accuracy of 70.6%

APPROACH OVERVIEW:

We use graph-based label propagation for cross-lingual knowledge transfer and use the projected labels as features in an unsupervised model

ALGORITHM 1:

CALCULATION OF TRIGRAM SIMILARITY:

Nirmal Surange & Dowlagar Suman

Building a bilingual similarity graph built from a sentence-aligned parallel corpus.
we use two types of vertices in our graph:

- on the foreign language side vertices correspond to trigram types(using individual words as the vertices throws away the context necessary to disambiguate),
- the vertices on the English side are individual word types.

Graph construction makes use of two similarity functions.

- The edge weights between the foreign language trigrams are computed to indicate how syntactically similar the middle words of the connected trigrams are
- second similarity function, which leverages standard unsupervised word alignment statistics

For each trigram type x2 x3 x4 in a sequence x1 x2 x3 x4 x5 , we count how many times that trigram type co-occurs with the different instantiations of each concept, and compute the point-wise mutual information (PMI) between the two.

Given this similarity function, we define a nearest neighbor We use N(u) to denote the neighborhood of vertex u, and fixed n = 5 in our experiments.

To define a similarity function between the English and the foreign vertices, we rely on high confidence word alignments. Since our graph is built from a parallel corpus, we can use standard word alignment techniques to align the English sentences and their foreign language translations
Based on these high-confidence alignments we can extract tuples of the form [u ↔ v], where u is a foreign trigram type, whose middle word aligns to an English word type v. Our bilingual similarity function then sets the edge weights in proportion to these tuple counts.

POS PROJECTION:

we run a single step of label propagation, which transfers the label distributions from the English vertices to the connected foreign language vertices.

<u>IMPLEMENTATION:</u>

1. TOKENIZATION:
    1. Given Hindi(Tagged) and Marathi(Untagged) parallel data, we (preprocessed) tokenized the languages so as to remove all the invalid symbols in data.
2. EXTRACTING ALL THE UNIGRAMS,BIGRAMS,TRIGRAMS AND 5-GRAMS:
    1. For Marathi, As the POS tags must depend upon the context we have used the left word and right word as the context for the data (which are trigrams)
        - Ex: x2 x3 x4
    2. Bigrams , 5 grams are the features used so as to calculate the edge weights between two trigrams.
    3. 5-grams are taken as the context to the given trigrams.
        - Ex: x1 x2 x3 x4 x5
    4. Bigrams are taken as left and right corner words for the given trigram with context. (ie., 5 gram)

- Ex: Left corner word for center word x3 : x1 x2
- Right corner word for center word x3 : x4 x5

5. To extract left and right context for each word, first we padded the data so that the corner words must not be left out.
6. We extracted all the N-grams

3. CALCULATION OF PMI VALUES:

1. For all N-grams we have calculated the Point wise Mutual Information, which says , Measure of how much one word tells us about the other. PMI is given as:

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

4. SUMMING UP THE FEATURE VALUES:

1. For each trigram taken we have found out all of its 5-grams ie., for every 5-gram word we searched whether the given trigram occurs in the center of it. If we get a hit then we stored the obtained 5 gram word and its PMI score.
2. For the obtained 5 gram word we took its left context(x1 x2) and right context (x4 x5) and we stored the right context and left context PMI values.
3. All these are done as Feature extraction for the given trigram word.(x2 x3 x4)
4. The same process is repeated for all the trigram words.

5. SIMILARITY FUNCTION:(A trigram by trigram matrix)

1. Now the data we contain is For each trigram word, we have its 5 gram word and its PMI score and from the 5-gram word we have the 2 bi-grams and their PMI score as features for the trigram words.
2. Similarity function says, how much the two trigrams are similar to one another using the features given above.
3. For the given two trigrams words, so as to calculate the similarity, we check whether the two trigram words have the same context ie., Is (x1....x5) are similar for the two words, if they are similar then we sum up the PMI scores of the 5 grams of the two words, the same is repeated for its other features (either bigrams or 5grams which are the features for the given trigram word), when there is a feature match, all are summed up.
    - The similarity will be of the form trigram1_trigram2 = (Summed up) PMI score
4. The above step is repeated for all the trigram words.
5. The matrix we obtained is the weighted graph which has the marathi trigrams as the vertices and the edges have the PMI score as its weights.
6. The matrix is a very sparse matrix
7. The similarity function says that some words behave similarly in a given context. If the edge weight is more, most similar behaviour, If the edge weight is zero, no similar behavior.

6. GIZA WORD ALIGNMENT:

1. 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.
2. First Giza is run taking Hindi tagged data as source and Marathi tagged data as target
3. Next Giza is run taking Hindi tagged data as target and Marathi tagged data as source.
4. The intersection of the above two alignments is taken.

7. EXTRACTION OF TAGS FROM THE GIZA++ ALIGNMENTS:

1. From the cross alignment matrix which have data as the form (Marathi_word#Hindi_word/Tag), we extract the data of the form (Marathi_word Tag)
2. As Giza++ gives the word alignments, syntactically, Marathi word will be of the same category of the given Hindi word
3. We will get a file containing Marathi_word, tag and its count, count says how many times the Marathi words with its particular tag may occurred in the corpus.
4. One Marathi word can have more than one tag.

8. 5 NEIGHBORS CALCUATION:

1. As the similarity matrix says more about the center word,
2. Given the similarity matrix we extract the center word from each trigram word, it will be of the form like
   - trigram1_trigram2 = (Summed up) PMI score => centerword1_centerword2 = (Summed up) PMI score
3. For each word ie., center word we extract the center words which have maximum PMI scores. These words form the 5 neighbors for the center word in the graph.
4. This calculation is done so because,(in the cross alignment matrix some words will be having no high confidence alignments, so those will have no tags) if a word has no tag aligned to it, it can take the tag from one of its neighbors.

9. RI COUNT:

1. This gives the tag distribution for the given center words.
2. 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.

   Calculated using the formula:

$$\mathrm{r}_i(y) = \frac{\sum_{v_y} \#[u_i \leftrightarrow v_y]}{\sum_{y'} \sum_{v_{y'}} \#[u_i \leftrightarrow v_{y'}]}$$

10. QI COUNT:

1. This gives the distribution of word for all the tags.
2. 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.

Calculated using the formula:

$$\mathrm{q}_i^{(m)}(y) = \begin{cases} \mathrm{r}_i(y) & \text{if } u_i \in V_f^l \\ \dfrac{\gamma_i(y)}{\kappa_i} & \text{otherwise} \end{cases}$$

$$\gamma_i(y) = \sum_{u_j \in \mathcal{N}(u_i)} w_{ij} \mathrm{q}_j^{(m-1)}(y) + \nu\, U(y)$$

$$\kappa_i = \nu + \sum_{u_j \in \mathcal{N}(u_i)} w_{ij}$$

11. POS INDUCTION:

1. 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.
2. Now we have the tagged Marathi data as result.

RESULTS:

For the given input of the form:

स्वच्छ उच्छ्वास आणि दात हे आपले व्यक्तिमत्व खुलवतील .
दातांमुळे आपला आत्मविश्वाससुद्धा वाढतो .
आपल्या हिरड्यांच्या आणि दातांच्यामध्ये जीवाणू असतात .
हे दातांना अस्वच्छ आणि श्वासांना दुर्गंधित करतात .
इथे दिलेल्या काही सोप्या सूचना मदतीने आपण आपल्या दातांना आणि उच्छ्वासास स्वच्छ ठेवू शकतो .
दातांना नीट साफ करा .
दातांना नीट साफ करण्यासाठी दोन ते तीन मिनटांचा कालावधी लागतो .
परंतु बहुतेक लोक ह्याच्यासाठी एक मिनटापेक्षाही कमी वेळ देतात .
खूप पाणी प्या .
तोंड कोरडे पडल्यावर जीवाणू जोरात हल्ला करतात .

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
आपल्या_PR हिरड्यांच्या_N आणि_CC दातांच्यामध्ये_N जीवाणू_N असतात_V
हे_DM दातांना_N अस्वच्छ_JJ आणि_CC श्वासांना_N दुर्गंधित_JJ करतात_V
इथे_DM दिलेल्या_V काही_QT सोप्या_JJ सूचना_N मदतीने_N आपण_PR आपल्या_PR दातांना_N आणि_CC उच्छ्वासास_N स्वच्छ_JJ ठेवू_V शकतो_V
दातांना_N नीट_JJ साफ_N करा_V
दातांना_N नीट_JJ साफ_N करण्यासाठी_V दोन_QT ते_PSP तीन_QT मिनटांचा_N कालावधी_N लागतो_V
परंतु_CC बहुतेक_QT लोक_N ह्याच्यासाठी_PR एक_QT मिनटापेक्षाही_RP कमी_QT वेळ_N देतात_V

खूप_RP पाणी_N प्या_V
तोंड_N कोरडे_JJ पडल्यावर_V जीवाणू_N जोरात_RB हल्ला_N करतात_V

EXPERIMENTS CONDUCTED & ACCURACY:

**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.

**As you can see that, the data is not BIS tagged, because the given Hindi data is very asynchronous, it is not preprocessed correctly, we are unable to extract all the BIS tags form the Hindi data, even if we extract them, we are getting 96 tags, this is so because, as the data is not preprocessed, we find some tags are mixed with the UTF words. So from the BIS tagged hindi data which is of the format(N_NNP) we took only (N) tag.**