CS 545 PROJECT REPORT

Project Title : Richer Sequence Models for morphology

Members : Kumaresh Visakan, Murugan

Paul Suganthan, Gnanaprakash Christopher

Deepika, Muthukumar

Abstract

The current state-of-the-art part of speech taggers ignore the morpho-syntactic information. Morpho-syntactic taggers exist for highly inflectional languages like Arabic, Slovene etc. But no significant work has been done to label morpho-syntactic information independent of language. Morpho-syntactic information includes attributes like gender, case, number, tense etc. We propose a richer sequence model which labels the tags along with their morpho-syntactic information. We use the Multext-East morpho-syntactic descriptions for training the model. The Multext-East corpora contains morpho-syntactic annotation for many languages. The model uses features relevant to each morpho-syntactic attributes like gender,case etc. The attributes like gender, case etc have their own dependencies. Thus the selection of suitable features that model all the attributes, tends to be a hard problem. We learn the weights for features using an averaged perceptron.

The traditional Linear Sequence Model performs reasonably better in predicting the morpho-syntactic tag sequence. But its running time increases enormously with the increasing number of morpho-syntactic tags across languages. We propose three other models to optimize the traditional linear sequence model.

We evaluate the proposed models over various languages. Also, we compare the models with a HMM and traditional sequence model with emission and transition features.

Introduction

Morpho-syntactic annotation includes tagging attributes like gender, case, number, tense etc along with POS tag for the word. For example, the following English sentence is annotated with morpho-syntactic tags.

It\Pp3ns was\Vmis3s a\Di bright\Af cold\Afp day\Ncns in\Sp April\Ncns ,\PUN and\Cc-n the\Dd clocks\Ncnp were\Vais-p striking\Vmpp thirteen\Mc .\PUN

Thus there are many possible morpho-syntactic tags in each language. The total number of MSD(morpho-syntactic description) tags in each language is shown in the table below.

Language	**Number of MSD tags**
Bulgarian	338
Czech	1425
English	135
Estonian	642
Farsi	428
Hungarian	17279
Polish	1324
Romanian	616
Slovak	1612
Slovene	1902
Serbian	1243

Data Set :

We use the Multext East annotated corpora. Multext East corpora contains morpho-syntactic annotated text for Bulgarian, Czech, English, Estonian, Farsi, Hungarian, Polish, Romanian, Serbian, Slovak, Slovene. It contains the morpho-syntactic specifications and annotated "1984" corpus for each language.

Training Data : We use the first 5500 sentences of the "1984" corpus as training data.

Test Data: We use the remaining sentences of the "1984" corpus as test data.

Related Work

There hasn't been any significant work done on language independent morpho-syntactic tagging. Morpho-syntactic tagging has been only applied to inflectional languages like German, Arabic, Slovene etc. Language specific morpho-syntactic taggers has been developed for Slovene [1], Arabic [3] and Croatian [2]. Building a language specific morpho-syntactic tagger utilizes the structural properties of that language. Whereas there hasn't been any state-of-the-art morpho-syntactic tagger, which is language independent.

Morpho-Syntactic Tagging Models

Model 1(Baseline) : Bigram HMM with transition and emission features

We implemented a bigram HMM. An HMM is defined by transition parameters $P(t|t')$ and emission parameters $P(w|t)$, which form a set of multinomial distributions. We will estimate these parameters using Maximum Likelihood over the training data. The estimates will take the following forms,

$$P(t|t') = \frac{count(t', t)}{count(t')}$$

$$P(w|t) = \frac{count(t, w) + \delta}{count(t) + |V|.\delta}$$

where $\delta = 0.000001$

$|V|$ is the number of word types observed across the whole file, including training and test sentences.

We are smoothing the emission parameters to account for unseen words.

Once we have estimated these parameters, we can now predict part-of-speech tag sequences for the test sentences using the Viterbi algorithm. Viterbi algorithm computes $max_t\ P(w|t)$ .By keeping track of backpointers, we can simultaneously output the best tag sequence:

$$t^* = \arg max_t\ P(t|w) = \arg max_t\ P(w, t)$$

We consider this model to be the baseline. We thereby compute overall accuracy and unseen accuracy of this model for all the 11 languages. The table below, shows the accuracy of each language using the baseline.

Language	Overall Accuracy	Unseen Accuracy
Bulgarian	84.55	10.69
Czech	73.38	4.81
English	89.73	6.34
Estonian	81.04	3.65
Farsi	86.85	6.05
Hungarian	79.67	1.00
Polish	70.29	2.25
Romanian	83.68	6.10
Slovak	72.66	3.77
Slovene	78.39	3.82
Serbian	73.2	3.24

Model 2: Modified HMM with higher smoothing for most common tag

Now we will try improve the HMM performance on unseen words. The key idea is to bias our model so that it is more likely to use the overall most frequent tag when a word is unobserved in training. If our model had direct parameters for tags given words, $P(t|w)$, we could do so easily using a non-symmetric Dirichlet prior – by using a higher hyperparameter value for the dimension corresponding to the most common tag. Operationally, this would consist of using a higher smoothing pseudo-count for the most frequent tag than for the other tags.

Unfortunately, the HMM model goes in the reverse direction, and is parameterized by $P(w|t)$ (word given tag), so an asymmetric smoothing prior is not an option. An indirect way of encouraging the use of the most frequent tag for unseen words is to use a larger smoothing value for the $P(w|t_{freq})$ distribution than for the other emission distributions (where t_{freq} refers to the most frequent overall tag). In this way, more probability mass will be reserved for unseen words emitted from this tag than from the other, less common tags.

To implement this, we re-estimate the HMM and rerun Viterbi with the following smoothed estimator:

$$P(w|t_{freq}) = \frac{count(t_{freq}, w) + 1000\delta}{count(t) + |V|.1000.\delta}$$

$$P(w|t) = \frac{count(t, w) + \delta}{count(t) + |V|.\delta}, \forall t \neq t_{freq}$$

The table below, shows the accuracy of each languages using the modified HMM.

Language	Overall Accuracy	Unseen Accuracy
Bulgarian	83.79	5.95
Czech	73.11	2.65
English	91.55	35.76
Estonian	81.03	1.52
Farsi	87.98	27.05
Hungarian	79.55	0.33
Polish	70.11	0.85
Romanian	83.57	4.72
Slovak	72.52	2.24
Slovene	78.13	2.08
Serbian	73.22	2.31

The unseen accuracy of English and Farsi show significant improvement from the previous model. For all other languages, the accuracy decreases slightly. This is due to the fact that in most of the languages, the most frequent tag is 'PUN'. Where as the most frequent tag in Farsi is 'Nc-s' and in English is 'Ncns'. Since, the modified HMM gives a higher smoothing for the most frequenct tag, only English and Farsi show significant improvement in unseen accuracy.

Model 3 : Feature based prediction model

In this model, we use a feature based model to predict the tags. We use 2 types of features.

Emission feature :

For each tag-word pair (t, w), we define the following emission feature

$$f(w_i, t_i) = \begin{cases} 1\,, if\ t_i = t\ and\ w_i = w \\ 0\,, otherwise \end{cases}$$

Suffix features :

We add the following suffix features,

- w_i ends with character x , $length(w_i) > 1$, and $t_i = t$
- w_i ends with characters xy , $length(w_i) > 2$, and $t_i = t$
- w_i ends with characters xyz , $length(w_i) > 3$, and $t_i = t$

Where $x, y\ and\ z$ range over all possible triples of characters, and t ranges over all possible tags

This model tags each sentence, incrementally tagging each word in the sentence. We don't have to use Viterbi algorithm to predict the tags, since we don't use transition features.

The table below, shows the accuracy of each language using the feature based prediction model.

Language	**Overall Accuracy**	**Unseen Accuracy**
Bulgarian	88.88	56.51
Czech	81.17	52.01
English	89.84	65.94
Estonian	87.47	62.05
Farsi	86.06	49.04
Hungarian	91.40	63.18
Polish	78.64	47.32
Romanian	90.33	60.59
Slovak	79.23	49.00
Slovene	84.19	47.76
Serbian	77.96	43.29

All the languages show significant improvement in overall accuracy and unseen accuracy when compared to the HMM models.

Model 4 : Linear Sequence model with naive Viterbi

Now we will consider a discriminative linear sequence model. This model scores each sentence w along with a potential tagging , in the following way:

$$score(w, t) = \lambda . F(w, t)$$

where $F(w, t)$ is a vector-valued global feature function, and λ is a corresponding weight vector. In order to use the Viterbi dynamic programming algorithm, we have to ensure that the global feature function decomposes along the positions of the sentence in the following way,

$$F(w, t) = \sum_i f(i, w, t_i, t_{i-1})$$

where i ranges over the indices of the sentence. We will refer to $f(i, w, t_i, t_{i-1})$ as a local feature function. It will be of the same dimensionality as the global feature function, and will typically consist of binary values indicating the presence or absence of individual features at position i in the tagged sentence.

We use 3 types of features. They are transition feature, emission feature and suffix features. The emission feature and suffix features are similar to features used in feature based prediction model.

Once we have the weight vector λ , we can use Viterbi algorithm to predict the tags. We learn the weights using a averaged perceptron. We use 5 rounds of training for the averaged perceptron. The averaged weight vector is of the form,

$$\lambda = \frac{1}{T}[\lambda^{(1)} + \cdots + \lambda^{(T)}]$$

where $\lambda^{(i)}$ is the weight vector obtained after i_{th} iteration,

$$T = number\ of\ iterations * number\ of\ training\ sentences$$

This model considers the whole morpho-syntactic annotation as a single tag. This model uses Viterbi algorithm to predict the morpho-syntactic tag sequence. Since the number of morpho-syntactic tags in each language is large, this model takes significant amount of running time.

The table below, shows the accuracy of each language using this model.

Language	**Overall Accuracy**	**Unseen Accuracy**
Bulgarian	91.43	71.60
Czech	87.38	65.39
English	94.96	77.17
Estonian	90.92	73.57
Farsi	88.34	58.11
Hungarian	89.53	66.45
Polish	83.21	55.65
Romanian	94.79	75.65
Slovak	83.26	57.37
Slovene	87.04	56.24
Serbian	80.54	53.50

All the languages show significant improvement in overall accuracy and unseen accuracy when compared to the previous models.

Model 5 : Linear Sequence Model with Greedy Viterbi heuristic

This model uses a discriminative linear sequence model considering the whole morpho-syntactic annotation as a single tag, with a reduced search space. This model uses a heuristic similar to beam search. We try to optimally reduce the search space of Viterbi algorithm. For

each word in test sentence we only consider the transition between tags that occur in training data.

The table below, shows the accuracy of each language using this model.

Language	Overall Accuracy	Unseen Accuracy
Bulgarian	92.33	71.50
Czech	84.56	62.35
English	94.39	77.45
Estonian	89.89	70.51
Farsi	88.93	58.11
Hungarian	90.70	67.69
Polish	82.08	53.52
Romanian	93.50	77.01
Slovak	82.19	56.55
Slovene	86.26	54.87
Serbian	82.53	55.28

The accuracy almost remains same for most of the languages compared to the previous model, but with a significant reduction in running time. This model behaves as naive Viterbi model, when all possible morpho-syntactic tag transitions occur in training data.

Model 6: Linear sequence model with Single Phase Viterbi

For each sentence, we predict the main POS tag sequence using Viterbi algorithm. Once the main POS tag sequence for a sentence is fixed, we predict the morpho-syntactic tag, by considering all the possible morpho-syntactic tags based on the main POS tag. The prediction of morpho-syntactic tag is similar to the feature based prediction model, but considers candidate morph-syntactic tags based on the main POS tag. Thus the morpho-syntactic tag with highest weight among the possible choices is chosen. Here also, we don't use the transition feature while predicting morpho-syntactic tag. We use transition feature only for predicting main POS tag.

The table below, shows the accuracy of each language using this model .

Language	Overall Accuracy	Unseen Accuracy
Bulgarian	90.09	65.69
Czech	81.14	53.71
English	91.74	71.74
Estonian	87.63	64.11

Farsi	86.62	51.98
Hungarian	91.84	66.71
Polish	77.92	47.36
Romanian	92.02	66.82
Slovak	79.72	50.75
Slovene	83.59	49.05
Serbian	78.17	45.39

All the languages show significant improvement in accuracy compared to the HMM models and feature based prediction model. The accuracy decreases slightly for most of the languages compared to the Naïve Viterbi and Greedy Viterbi models. But this model shows significant reduction in running time compared to previous models.

Model 7: Linear Sequence Model with 2-Phase Viterbi

This model is similar to the previous model except the fact that this model uses Viterbi algorithm in two phases, initially to predict the main POS tag sequence, and then to predict the full morpho-syntactic tag. The previous model doesn't use Viterbi to predict the morpho-syntactic tag.

In this model, for each sentence we first predict the main POS tag sequence using Viterbi algorithm. We use transition, emission and suffix features to predict the main POS tag sequence. The morpho-syntactic tag for each word, is then predicted by employing Viterbi in reduced search space comprising of transitions between morpho-syntactic tags based on the POS tag of previous word, to the morpho-syntactic tags based on the POS tag of the current word.

The table below, shows the accuracy of each language using the 2-Phase Viterbi model.

Language	**Overall Accuracy**	**Unseen Accuracy**
Bulgarian	90.47	67.56
Czech	86.82	63.65
English	93.60	73.82
Estonian	89.66	65.91
Farsi	88.17	51.98
Hungarian	91.36	67.58
Polish	84.60	55.06
Romanian	91.82	68.97
Slovak	84.04	58.30
Slovene	87.99	56.55
Serbian	83.18	53.14

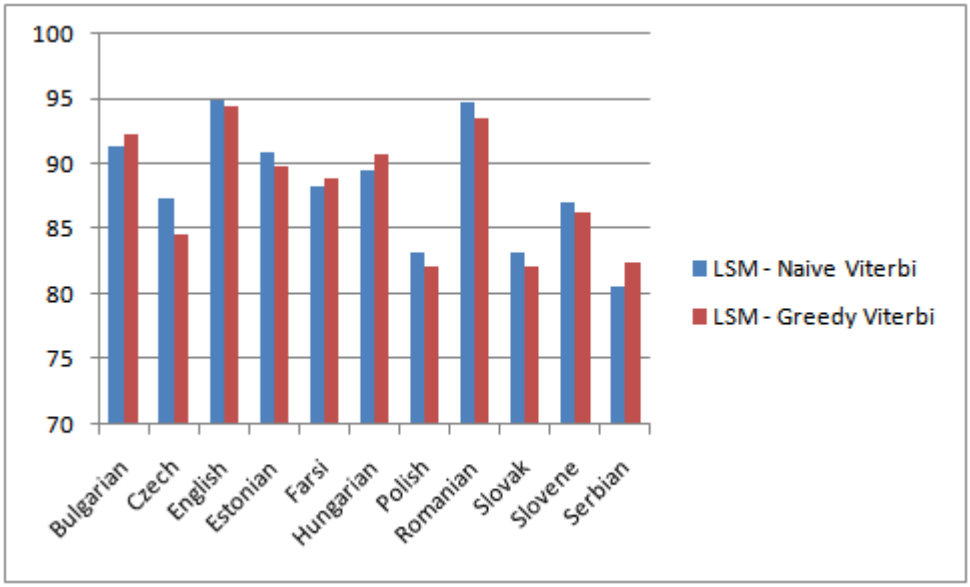
The accuracy increases slightly compared to the previous model. But the accuracy of most languages are almost similar compared to the naive Viterbi and Greedy Viterbi models.

Analysis

Among all the models, LSM with naive Viterbi, LSM with 2-Phase Viterbi and LSM with Greedy Viterbi, perform better than other models. But LSM with naive Viterbi model has constraints on running time. Also LSM with Greedy Viterbi model, tends to behave like naive Viterbi model as the number of possible morpho-syntactic tag transitions increases in training data. **Thus LSM with 2-Phase Viterbi performs reasonably better, with reduced running time**.

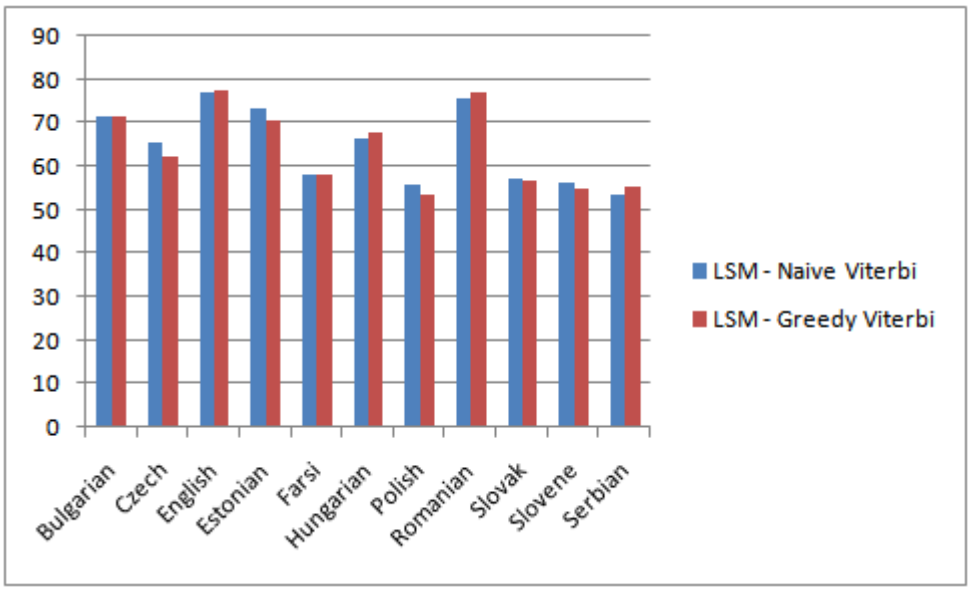
Comparison of LSM(with Naive Viterbi) and LSM(with Greedy Viterbi) :

The following bar graph compares the overall accuracy of LSM(with Naive Viterbi) and LSM (with Greedy Viterbi) , for all languages.



Both the models perform comparatively better. For some languages, LSM(with Naive Viterbi) performs slightly better whereas for other languages, LSM(with Greedy Viterbi) performs slightly better.

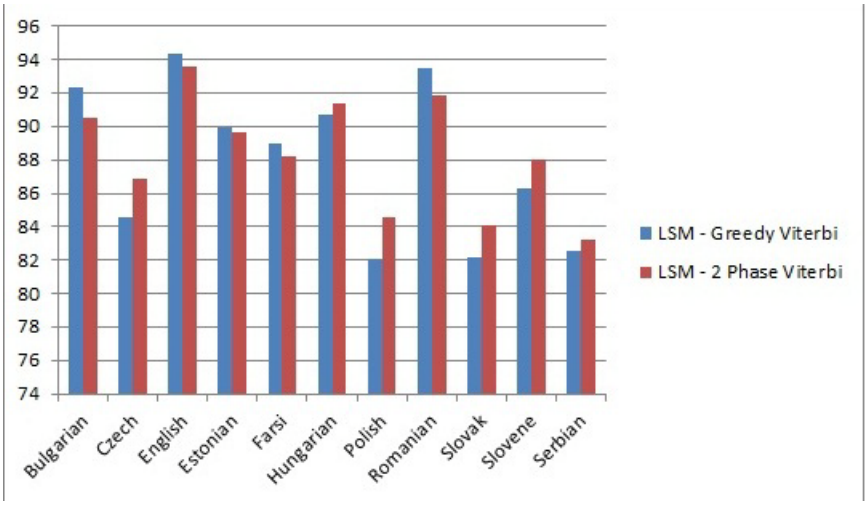
The following bar graph compares the unseen accuracy of LSM (with Naive Viterbi) and LSM (with Greedy Viterbi), for all languages.



Both the models perform comparatively better. For some languages, LSM(with Naive Viterbi) performs slightly better whereas for other languages, LSM(with Greedy Viterbi) performs slightly better.

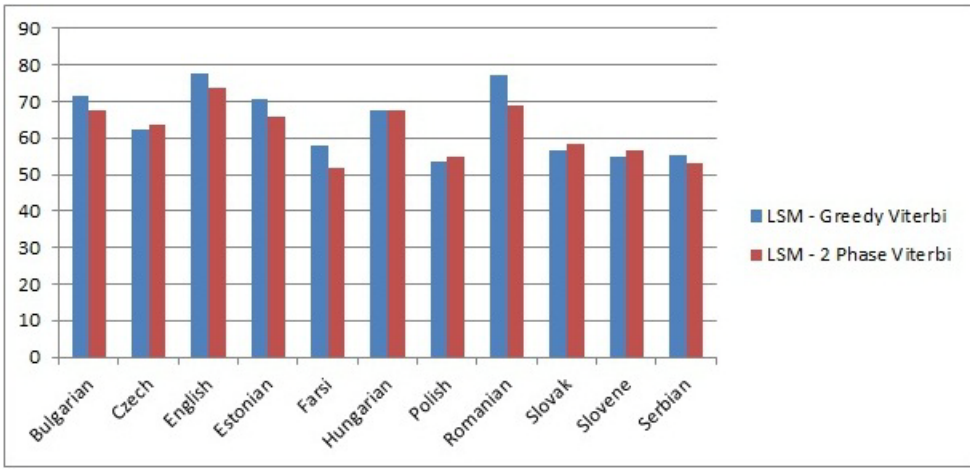
Comparison of LSM(with 2-Phase Viterbi) and LSM(with Greedy Viterbi) :

The following bar graph compares the overall accuracy of LSM(with 2-Phase Viterbi) and LSM (with Greedy Viterbi) , for all languages.



Both the models perform comparatively better. For some languages, LSM(with 2-Phase Viterbi) performs slightly better whereas for other languages, LSM(with Greedy Viterbi) performs slightly better.

The following bar graph compares the unseen accuracy of LSM (with 2-Phase Viterbi) and LSM (with Greedy Viterbi), for all languages.



Both the models perform comparatively better. For some languages, LSM(with 2-Phase Viterbi) performs slightly better whereas for other languages, LSM(with Greedy Viterbi) performs slightly better.

Future Work

The selection of language dependent features could bring significant improvement in accuracy. Thus the study of language dependent features is a possible direction for future work. Also studying the effect of trigram models, on the performance of 2-Phase Viterbi and Greedy Viterbi is a possible direction for future work. Also, we are working on n-Phase Viterbi model, in which each individual morpho-syntactic attribute is predicted in each phase by fixing previously predicted attributes.

References

[1] Morphosyntactic Tagging of Slovene Legal Language. Informatica 30:483–488 (2006)

[2] Tagset Reductions in Morphosyntactic Tagging of Croatian Texts

[3] Morpho-Syntactic Tagging System Based on the Patterns Words for Arabic Texts