Compression between probabilistic models for segmentation and labeling sequence data

Brian Rikshpun

Guy Cohen

Ester Vaknin

Almog Klein

Afeka College of Engineering

October 17, 2021

**1    Abstract**

In this paper we studied two statistical classification models, one is a conditional random field - CRF which is a discriminative model and the other is the hidden Markov model - HMM which is a generative model. The purpose of our work is to compare between two statistical classification methods. The research question examined in this paper is how the CRF and hmm models will predict segmentation and labeling sequence data, trained with the same dataset.

We also study the effect of the training and test sizes on the accuracy of each model and the effect of the training optimization function on the CRF dependent on the train size.

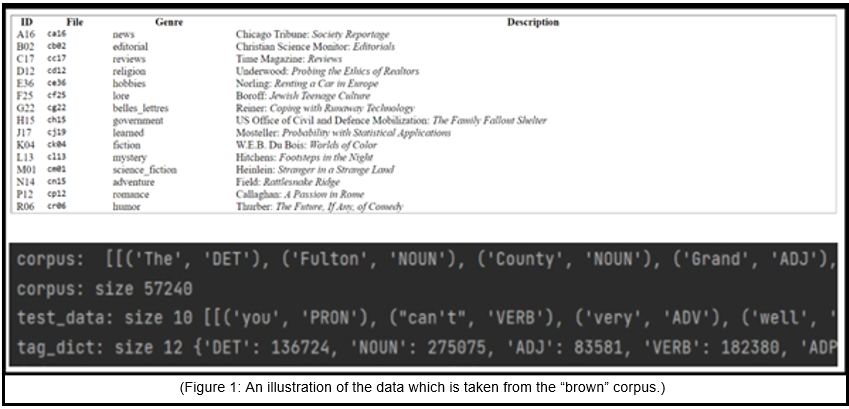
Data labeling has many applications such: chat bots, information extraction and completion of sentences and even sentiment classification of comments on movies/shows.

Today the ability to tag data sequences is an important and very necessary capability for applications in many areas such as NLP and image processing. Segmentation and labeling of data sequences gives the machine the ability to understand the relationships between database records.

The main challenges we faced in this work were finding a suitable code for the CRF model - which used more in tagging objects in images. We also found it hard to explain why the CRF has good accuracy on small test data as shown in our results [1].

**2    Data description**

In our work we used the “Brown” corpus, a present-day American English corpus. The Corpus was the first computer-readable general corpus of texts prepared for linguistic research on modern English. This corpus contains text from 500 sources, and the sources have been categorized by genre such as “news”, “reviews”, etc. In our work we didn't use specific categories. We imported the corpus by using the [NLTK - Natural Language](https://www.nltk.org/) Toolkit which contains all kinds of corpuses.



**3    Methodology of work**

The code that was implemented in our work was originally taken from "[GitHub](https://github.com/arnab39/POS-Tagging_HMM_vs_CRF/blob/master/POSTagging_HMM_vs_CRF.ipynb)", adjustments and changes were made in order to yield the results required to achieve the goals in our work. In addition, the code was originally implemented with the help of Jupiter notebook, but we chose to implement it in the "pycharm" environment for a better understanding of the model processes and how they work.

Our project GitHub - [Link](https://github.com/BrianRikshpun/MSC_NLP)

The main libraries used in the code are:

nltk.corpus - Natural Language Toolkit corpus readers. The modules in this package provide functions that can be used to read corpus files in a variety of formats. With the help of this library we loaded our corpus.

nltk - This library helps us with the hidden Markov model. We will use the functions for the purpose of calculating the distribution of the probability and frequency of the words and tags.

sklearn.metrics - APIs for evaluating the quality of a model’s predictions. This library helped us obtain a revaluation of the model quality indicators.

sklearn\_crfsuite - A CRFsuite (python-crfsuite) wrapper which provides scikit-learn-compatible CRF estimator. providing the option to train the model on the train data and capability of prediction on test data.

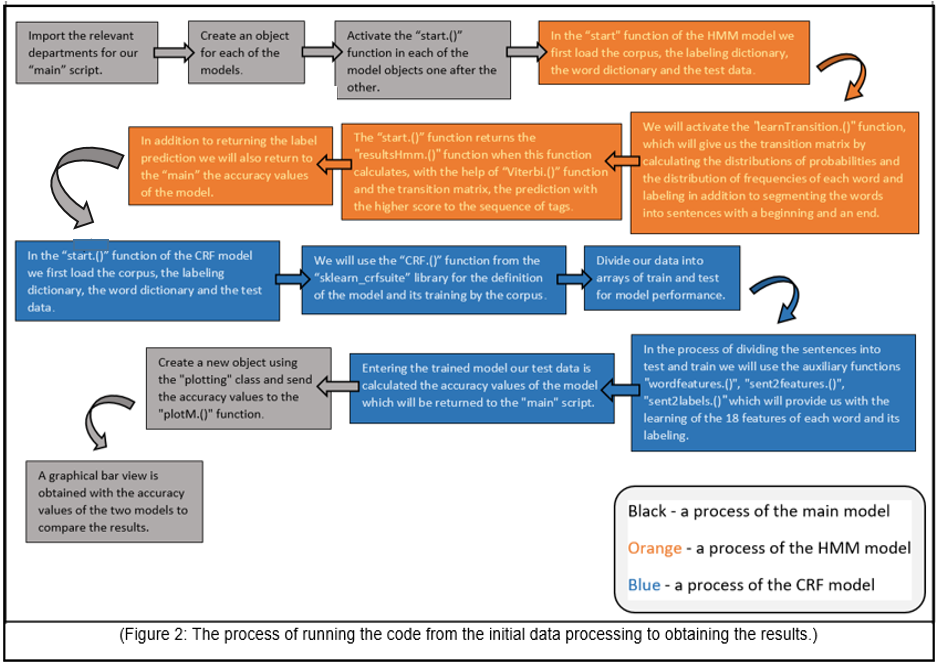
matplotlib.pyplot - A state-based interface to matplotlib. It provides a MATLAB-like way of plotting. Provided tools to display accuracy metrics graphically.

**Code work process**

Originally the code logic was one script running together as an ipynb file (Python notebook). We chose to split the code and build it as separate classes for a deeper understanding of the modeling algorithm.

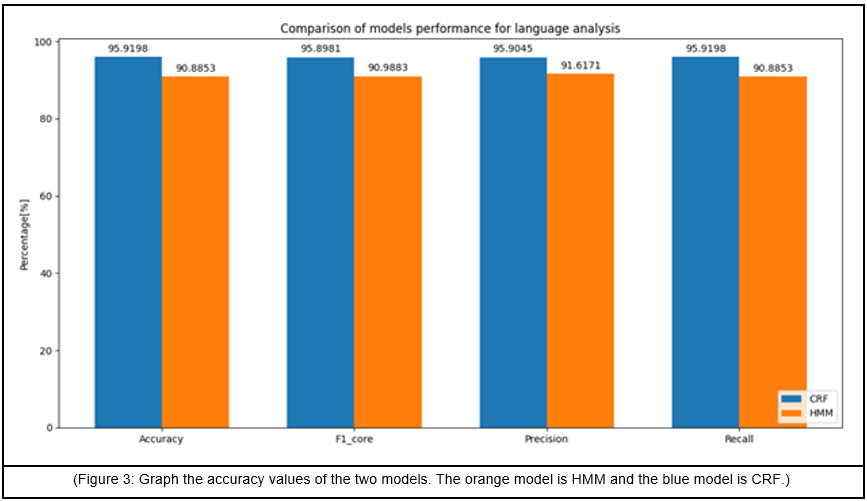
The code is divided into five scripts. The first is the “main” script which unites the other scripts together. The second script is called “logics” where we load our corpus for the two models and get from it the tagging dictionary and the word dictionary and in addition we take from the corpus new sentences for the test.

The next two scripts are the classes of the models, a third script for the class of the HMM model which is called “hmmModule” and a fourth script for the class of the CRF model which is called “crfModule”. The fifth and final script is called “plotting”, a class for displaying the results of the models graphically.



In order to examine the performance of the models, we chose to run the models on the whole corpus data which contains 57,370 sentences and 1,161,671 words, divided into 80% training and 20% test. The results are presented below (Figure 3). We are measuring and comparing between the models according to the Accuracy, Precision, Recall and F1 scores for each model.

**4    Results**

****

The graph gives us a clear picture for comparison between the models.

From the graph it can be clearly seen that for the same amount of train (45,872 sentences & 979,646 words) and for a test (11,498 sentences & 182,025 words). The CRF model achieves better results in terms of all indices compared to the HMM model. Yet the HMM model yields good predictive results relative to other statistical models.  A statistic that can explain the behavior of the models is the ratio of the identical words that exist between the test and the train. Based on the graph data, 174,856 identical words were found between the test data and the train data. This means that there is an overlap of 96% of the words present in the test also appear in the train. In addition, the CRF model creates broader connections between each word and its features [2] [3]. Which is reflected in higher prediction percentages as seen from the graph.

**5    Innovations**

Some changes and innovations to the code were made in our study:

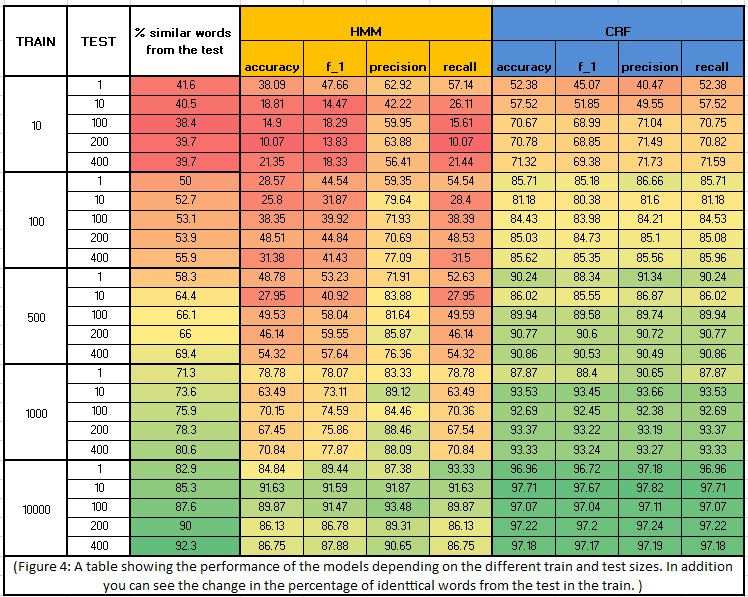
* Code Classes - Changes and adjustments have been made to the original code of this paper. Divided the code to main classes.
* Hyper Parameters - Experimenting few training algorithms (CRF hyper parameters). Adaptive Regularization of Weight Vector (arow), Passive Aggressive (pa), Averaged Perceptron (ap), Stochastic Gradient Descent with L2 regularization term (l2sgd) and Gradient descent (lbfgs).
* Training & Test sentences - In terms of changes made in the train and test data size aspects, we examined a number of different sizes in both aspects.

The number of sentences that were examined in the Train Data: 10, 100, 500, 1,000, and 10,000 sentences.

The number of sentences that were examined in the Test Data: 1, 10, 100, 200, and 400 sentences.

**6    Conclusions**

CRF is the model with the best accuracy results compared to HMM [3]. The CRF model calculates for each word its 18 features, in addition to calculating the tag probabilities according to the order of each word in the sentence. This model creates stronger connections between each word in the word dictionary and the different labels. And so in fact he overcomes the label Bayes problem easily [1]. In contrast, the HMM model creates connections only between a word and the tags & neighbor words [4]. However, we have seen that if there is a large percentage of word overlap between the test and the train data, this model will also yield relatively good results in the small train sets.



The significance of the ratio between the total words that exist in the test data and the total number of identical words between the test and train data is of great importance and directly affects the accuracy results of the models.

Based on this table it can be seen that already at a train size of 10 sentences we get a 40% overlap between the words in the test and the words in the train. These are the data which were taken from the “Brown” corpus and in fact this might explain the high percentages of the model given a small training dataset.

**7   Reference**

[1] Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data (2001), <http://portal.acm.org/citation.cfm?id=655813>

[2] Conditional Random Fields: An Introduction (2004),

<https://repository.upenn.edu/cgi/viewcontent.cgi?article=1011&context=cis_reports>

[3] The Label Bias Problem (2020),

<https://awnihannun.com/writing/label_bias.pdf>

[4] Hidden Markov Models (2004),

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.1016&rep=rep1&type=pdf>

[5] A Revealing Introduction to Hidden Markov Models (2021),

<http://www.cs.sjsu.edu/faculty/stamp/RUA/HMM.pdf>

[6] On the Effect of the Label Bias Problem in Part-Of-Speech Tagging (2013),

<https://ieeexplore.ieee.org/abstract/document/6719875/>

[7] A Maximum Entropy Model for Part-Of-Speech Tagging (1996),

<https://www.aclweb.org/anthology/W96-0213.pdf>