Proposed strategy 1

The model used till now was the n -gram approach.

One approach which can be implemented is the Markov model.

If a document is given, each word can be bifurcated as a Markov chain of letters.

When the entire document is taken into account as one complete set of Markov chains, the set of starting and transitional probabilities can be calculated and referred to as a Markov Model for that particular language.

This proposed model in our research project which will not only identify the languages with a lower error rate, but will also result in faster identification speed as compared to N-gram model.

The occurrences of letters in a word can be regarded as a stochastic process and hence the word can be represented as a Markov chain where letters are states. The occurrence of the first letter in the word is characterized by the initial probability of the Markov chain and the occurrence of the other letter given the occurrence of its previous letter is characterized by the transition probability.

**Updated Strategy 1:**

In order to define an HMM completely, following elements are needed.

* The number of states of the model, *N*.
* The number of observation symbols in the alphabet, *M*. If the observations are continuous then *M* is infinite.
* A set of state transition probabilities tex2html_wrap_inline2612 .

displaymath2614

where tex2html_wrap_inline2616 denotes the current state.  
Transition probabilities should satisfy the normal stochastic constraints,

displaymath2618

and

displaymath2620

* A probability distribution in each of the states, tex2html_wrap_inline2622 .

displaymath2624

where tex2html_wrap_inline2626 denotes the tex2html_wrap_inline2628 observation symbol in the alphabet, and tex2html_wrap_inline2630 the current parameter vector.  
Following stochastic constraints must be satisfied.

displaymath2632

and

displaymath2634

If the observations are continuous then we will have to use a continuous probability density function, instead of a set of discrete probabilities. In this case we specify the parameters of the probability density function. Usually the probability density is approximated by a weighted sum of *M* Gaussian distributions tex2html_wrap_inline2638 ,

displaymath2640

where,

displaymath2604

tex2html_wrap_inline2646 should satisfy the stochastic constrains,

displaymath2648

and

displaymath2650

* The initial state distribution, tex2html_wrap_inline2652 .  
  where,

displaymath2654

Therefore we can use the compact notation

displaymath2656

to denote an HMM with discrete probability distributions, while

displaymath2658

to denote one with continuous densities. .

During training one or more HMMs are created for each language L as shown in in for an excellent HMM tutorial).

The database, obtained from Riek, Mistretta and Morgan at Sanders [14], is a three language subset of a malespeech, five language Rome Laboratory (RL) database. The subset comprises the first session from each of three languages (Russian, German, and Chinese). From 15 to 20 read-speech messages per language are available, each spoken by a unique speaker. This database was processed in two ways: (1) using half of the messages for training and half for testing according to the Sanders convention and (2) using jackknifing. Some experiments also used an alternate form of training and testing: during training, one HMM was trained per speaker; during testing on message m from language L, the language of the message model (not including the model for message m) most likely to have produced the test speech was hypothesized. In this alternate mode, the system was actually finding the training speaker that matched the test speaker most closely? The third database employed was the 20 language CCITT database [7] first used for language ID by Sugiyama 161. male) are available. On average, each utterance is about eight seconds long. As these messages were recorded at language dependent sites, the 8 kHz, IRS filtered version of the database was used to insure uniform bandlimiting across languages. The CCITT database was processed us ing half of the messages for training and half for testing according to the Sugiyama convention.

Markov chains, named after [Andrey Markov](https://en.wikipedia.org/wiki/Andrey_Markov), are mathematical systems that hop from one "state" (a situation or set of values) to another. For example, if you made a Markov chain model of a baby's behavior, you might include "playing," "eating", "sleeping," and "crying" as states, which together with other behaviors could form a 'state space': a list of all possible states. In addition, on top of the state space, a Markov chain tells you the probabilitiy of hopping, or "transitioning," from one state to any other state---e.g., the chance that a baby currently playing will fall asleep in the next five minutes without crying first. We can minic this "stickyness" with a two-state Markov chain. When the Markov chain is in state "R", it has a 0.9 probability of staying put and a 0.1 chance of leaving for the "S" state. Likewise, "S" state has 0.9 probability of staying put and a 0.1 chance of transitioning to the "R" state.

References:

[1]<https://pdfs.semanticscholar.org/2bf0/8addb83f51befa8b4bc7ed16b54ed34018d0.pdf>

[2]<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.630&rep=rep1&type=pdf>

[3] <http://www.cs.princeton.edu/courses/archive/spr05/cos126/assignments/markov.html>

[4]

<http://jedlik.phy.bme.hu/~gerjanos/HMM/node4.html>

[5]

http://www1.cs.columbia.edu/~fadi/candidacy/LID/zissman93.pdf