International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 5, Number 1 (2015), pp. 11-18
© International Research Publications House
http://www. irphouse.com

Design a Model of Language Identification Tool

Safwan al Salaimeh[1], Zeyad Al Saraireh[2], Jawad Hammad Al Rawashdeh[3]

1,2 Computer Science Department, Jadara University.
3. Computer science Department, Al Majmaah University.

ABSTRACT

Language Identification tool is a system that give you the knowledge of what is the language that you are reading or write.
It is very essential for the user to know what the language that he/she dealing with. It is very useful for the user to know what the language he/she is dealing with to be easy editing for him or translate it. This system is helpful for the users that have issues with identify what language he/she dealing with.
The system will include various languages e.g. English, France, German, etc… and subjects for conducting identification. This system helps conducting to identify the language quickly and can thus help them to save time and the operations will carried out efficiently.
With the effective use, any user can apply the identification language tool and getting better results in less time.

Keywords: Identification tool, system, operation, conducting identification, effective use, results.

1. INTRODUCTION

The language identification TOOL (LIT) was built using PHP and My SQL
And the language identification models such that:

1- Unigram model
2- Biogram model
3- Trigram model

The system was built to support the following main areas:

When you deal with a different country or region, knowing the local language will help you to communicate and integrate with the local community.

If your partner, in-laws, relatives or friends speak a different language, learning that language will help you to communicate with them. It will also give you a better understanding of their culture and way of thinking.

If your work involves regular contact with speakers of foreign languages, being able to talk to them in their own languages will help you to communicate with them. It may also help you to make sales and to negotiate and secure contracts. Knowledge of foreign languages may also increase your chances of finding a new job, getting a promotion or a transfer overseas, or of going on foreign business trips.

1.2 PURPOSE

The purpose of language identification tool is identifying languages in an efficient manner and no time wasting for searching websites.

The main objective of language identification tool is to efficiently identifying the candidate text through a fully automated system that not only saves lot of time but also gives fast results.

For users they give their convenience and time and there is no need for using extra materials like: Dictionary, Google, etc.

1.3 Benefits

Easy to use for administrator and users.

Easy to conduct use and quick results.

No need to think about what kind of language used.

1.4 Scope

We can develop the program to be effective commercially and is used in machine translation tool, web development, businesses and hotels... etc.

2. INTRODUCTION TO LANGUAGE MODELS:-

A language model **(LM)** is a critical component for many applications, including speech recognition. Enormous effort has been spent on building and improving language models. Broadly speaking, this effort develops along two orthogonal directions: The first direction is to apply increasingly sophisticated estimation methods to a fixed training data set (corpus) to achieve better estimation. Examples include various interpolation and back off schemes for smoothing, variable length N-grams, vocabulary clustering, decision trees, probabilistic context free grammar, maximum entropy models, etc [l]. We can view these methods as trying to “squeeze out” more benefit from a fixed corpus. The second direction is to acquire more training data. However, automatically collecting and incorporating new training data is non-trivial, and there has been relatively little research in this direction. An example is a cache LM, which uses recent utterances as additional training data to create better N-gram estimates. The recent rapid development of the World Wide Web (WWW) makes it an extremely large and valuable data source. Just-in-time language modeling [2] submits previous user utterances as queries to WWW search engines, and uses the retrieved web pages as unigram adaptation data.

In this paper we propose a method based on language model for language identification.

2.1 Hidden Markov Models:

A Hidden Markov Model (HMM) consists of a set of internal states and a set of observable tokens. A run of a hidden Markov model generates a hidden state sequence s1,..., sT and a sequence of observable tokens a1,..., aT.

$$P(w_1 n) = \prod_{k=1}^{n} P(wk | w_1 k - 1)$$

Where:-

P (w1n):- probability of a string of word or letter.

Wk: - current word or letter.

Wk-1:-previous word or letter.

Applications of HMMs:

- Speech Recognition. The hidden states are word positions and the observable tokens are acoustic feature vectors.
- Part of speech tagging. The hidden states are the parts of speech (noun, verb, adjective, and so on).
- DNA sequence analysis. The hidden states might be protein secondary structure or a position in a homologous sequence.

2.2. n-gram model:-

The “n-gram” method usee letter or word n-grams. Representing the Frequency of occurrence of various n-letter combinations in a particular language. The language identification process can be divided into two phases: training and identification. A language identification model is trained for each targeted language.

In the training phase:-

Lists of words with a known language are presented as alphabetic strings. The frequency of Occurrence of Sequences of consecutive n letters is estimated from a large language specific training sample.

An N-gram model uses the previous N-1 words to predict the next one.

$$P(\text{wn} \mid \text{wn - N} + 1 \mid \text{wn - N} + 2 \ldots \text{wn -1}) = \frac{\text{count (wn - N} + 1 \mid \ldots\ldots \text{wn)}}{\text{Count (wn - N} + 1 \mid \ldots \text{wn -1)}}$$

2.2.1 Unigram Model :-

It’s every sequence of one element in a string of tokens, which are typically letters, syllables, or words; they are n-grams for n=1.

For example if we enter the word " languages ", the program will cut it in this form :

1-gram	Frequencies
l	1
a	2
n	1

g	2
u	1
e	1
s	1

Figure 2.1 shows the cutting of the word “language” in the Unigram and the frequencies for letter

2.2.2 Bigram Model:-

It is every sequence of two adjacent elements in a string of tokens, which are typically letters, syllables, or words; they are n-grams for n=2. The frequency distribution of bigrams in a string is commonly used for simple statistical analysis of text in many applications, including in computational linguistics, cryptography, speech recognition, and so on.

Gappy bigrams or skipping bigrams are word pairs which allow gaps (perhaps avoiding connecting words, or allowing some simulation of dependencies, as in a dependency grammar).

Head word bigrams are Gappy bigrams with an explicit dependency relationship.

Bigrams help provide the conditional probability of a token given the preceding token, when the relation of the conditional probability is applied:

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})}$$

That is, the probability $P()$ of a token W_n given the preceding token W_{n-1} is equal to the probability of their bigram, or the co-occurrence of the two tokens $P(W_{n-1}, W_n)$, divided by the probability of the preceding token.

If we enter the word “languages ", the program will cut it in a Bigrams case:-

2-gram	Frequencies
. l	1
la	1
an	1
ng	1
gu	1
ua	1
ag	1
ge	1
es	1
s.	

Figure 2.2 shows the cutting of the word “language” in the igram and the frequencies for letter

2.2.3 Trigrams model:-

Is a special case of the N-gram, where N is 3. They are often used in natural language processing for doing statistical analysis of texts.

Let #(w) be the number of times that the word w appears in a certain training corpus. Let #(w1w12) be the number of times that the pair of words w1‘w2 occurs and similarly for #(w1, w2, w3) for the triple of words w1, w2, w3.

Let N be the total number of word occurrences. A interpolated trigram model predicts the word w3 following a given pair w1, w2 as follows:-

$$p\left(w3 \mid w1, w2\right) = \frac{\text{count}(w1,\ w2,\ w3)}{\text{count }(w1, w2)}$$

If we enter the word “languages ", the program will cut it in a trigram-

2-gram	Frequencies
. la	1
lan	1
ang	1
ngu	1
gua	1
uag	1
age	1
gs.	1

Figure 2.3. shows the cutting of word “language” in the Trigram and the frequencies for the letter

Another Example:-

The sentence "the quick red fox jumps over the lazy brown dog" has the following word level trigrams:

the quick red
quick red fox
red fox jumps
fox jumps over
jumps over the
over the lazy
the lazy brown
lazy brown dog
the qui k_r
he_ uic _re
e_q ick red_qu ck_

3. EXPEREMENT

3.1 Data set

In our experiments, we focused on distinguish languages. The language identification models were trained and tested (gets from Wikipedia.) on our in-house name databases. Fifteen languages were used in the experiments:

Languages	Size
Finish	454kb
polish	486kb
Portuguese	481kb
Danish	513kb
German	461kb
Indonesian	438kb
Italian	484kb
French	494kb
English	442kb
Arabic	443kb
Persian	487kb
Romanian	484kb
Russian	478kb
Spanish	440kb
Turkish.	502

Figure 3.1 shows the languages that used in our work and the size of them

Languages	Trigram	Bigram	Unigram
Finish	92%	84%	10%
polish	100%	95%	0%
Portuguese	76%	67%	51%
Danish	98%	96%	9%
German	76%	67%	40%
Indonesian	98%	98%	94%
Italian	90%	99%	70%
French	98%	96%	0%
English	97%	99%	68%
Arabic	100%	100%	99%
Persian	100%	8%	100%
Romanian	100%	98%	0%
Russian	100%	100%	100%
Spanish	79%	86%	97%
Turkish.	96%	93%	2%

Figure 3.2 shows the proportion languages in (Unigram, Bigram, Trigram) and (which one is the best)

A test word database was independently collected and originally intended to test the performance of our recognition system. It was also used to evaluate the generalization capability of the language identification methods.

3.2 Evaluation

N-gram and decision free based methods were evaluated in the experiments. All the name databases were implored of short names since this is believed to be one of the most difficult tasks for language identification. The n-gram language identification models, including bigram trigram and their enhanced.

Counterparts, trained on our in-house Name databases for each of the four experimented Languages decision trees with the context length of four were trained for each letter on the same training databases. The first results, illustrated by Table I, were obtained on the training set to measure the learning performance of the proposed Approaches. The decision tree approach outperform the "-gram in term of the learning capability. The trigram method is better than the bigram method and the unigram method. Is clearly improved by the enhanced approach. In order to have reliable statistical information. The trigram requires a larger training set (particularly the enhanced trigram). Moreover, the trigram also winsome ~ a significant amount of memory that might be a problem in embedded systems.

3.3 Comparing n- gram models:

In this section, we explain n-gram models and propose n-gram models for tag suggestion method in tagging system.

3.3.1 N-gram Models:

N-gram models are widely used in statistical natural language processing [1]. An n-gram is a substring of n characters or a substring of n words in a given text. For instance, a bi-gram is 2-word substring of a text and a tri-gram is 3-word substring. If there is a sentence ‘Thank you very much’, bi-grams are {Thank, you}, {you, very}, and {very, much}. Trigrams are {Thank, you, very} and {you, very, much}. When a sentence consists of m words, it is possible to make m-gram at most. N-gram models are used in indexing method. Using trigram, a word ‘thank’ can be indexed by the, hank, and ankh. N-gram models are also used for prediction. This method calculates probabilities from corpus data. Given n-1 words, n-gram models predict which word would be next word based on calculated probabilities. For example, n-gram models predict next word of given ‘thank you very’ sentence based on previous bigram or trigram examples. It could be ‘much’.

Our program proved that the best model in the n-gram model is A trigram model, and then a bi-gram.

And the weakest model that may give inaccurate results is unigram model.

And this figure shows our result:-

- Effectiveness of the program in Trigram=93%
- Effectiveness of the program in Bigram=85%
- Effectiveness of the program in Unigram=49%

4. THE SOLUTION:

Firstly, we insert a group of sentences (training sentences) from fifteen languages into a program which divide each sentence to many parts according to three ways :

1- Unigram: divide each sentence according to one letter.
2- Bigram: divide each sentence according to two letters.
3- Trigram: divide each sentence according to three letters.

Then, we insert the result of them into database.
Finally, we use these equations to distinct language.

$$P(wn | wn - N + 1 | wn - N + 2 \ldots wn - 1) = \frac{\text{count}(wn - N + 1 | \ldots\ldots wn)}{\text{Count}(wn - N + 1 | \ldots wn - 1)}$$

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})}$$

$$p(w3 | w1, w2) = \frac{\text{count}(w1, w2, w3)}{\text{count}(w1, w2)}$$

5. CONCLUSION

One thing is for sure, language identification is not simple. To achieve language identification will require further experimentation. Here in our program, we use N-gram for distinct language.

N-gram depends on the ranks which start from 1 to n, we use just the first three ranks which known as:

Unigram, Bigram, Trigram. As a result, the rank-order statistics method is not appropriate for short strings because rank ordering and sorting is slow and requires long strings.

After collecting a lot of sentences for each language from Web which sample size for each language ranges from 400-500 KB, we entered it in the program which was cited it to more parts according to the previous way, we entered these parts into database. Then we used some equations to distinct language for the entered sentence.

Finally, we inserted one hundred sentences for each language and make testing for it to measure the efficiency of our program depending on Unigram, Bigram, and Trigram separately and we found tri-gram is better than uni and bi-gram.

6. REFERENCES:

[1] www.ieee.org

[1.1] I. Schmitt, "Trigram-based Method of Language Identification", U.S. Potent. Number 5062143, October 1991.

[2] www.wikipedia.org

[3] D. Jurafsky and J. H. Martin, Speech and Language Processing: Pearson Education International, 2008.

[4] R. N. Moll, M. A. Arbib, and A. J. Kfoury, An Introduction to Formal Language Theory. New York: Springer-Verlag, 1988.