Predicting Correctness of “Google Translate”

Yulia Rossikova, J. Jenny Li, and Patricia Morreale
Computer Science, Kean University
1000 Morris Ave, Union NJ 08340 USA
juli@kean.edu

***Abstract*— This paper presents a new modeless approach for Machine Learning predictions, called Radius of Neighbors (RN). We applied RN to predict the correctness of Google translator and found it to be an improvement over K-Nearest Neighbors (KNN) in terms of prediction accuracy. Both methods are applicable to situations when a mathematical prediction model does not exist or is unknown. With RN, we will be able to create new applications that rely on the users’ awareness of translation accuracy, e.g. an online instant messager, which allows users to chat in various natural languages.**

Keywords – K-Nearest Neighbors (KNN), Machine Learning, Prediction.

EXTENDED ABSTRACT

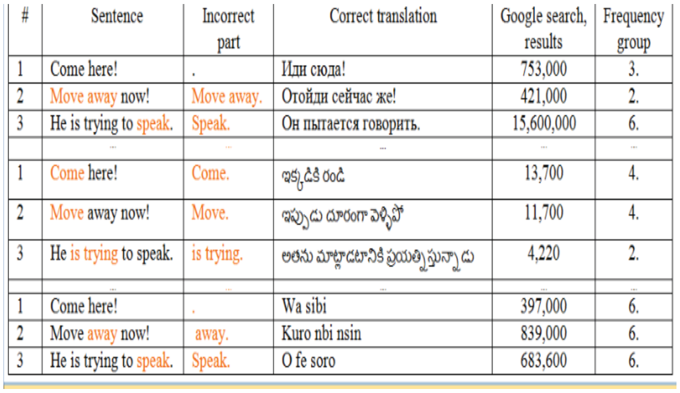
Machine learning in recent years has given us many new breakthrough applications such as self-driving cars, phone contact center voice recognision, effective web search and automatic natural language translators. Prediction is a key feature of machine learning and K-Nearest-Neighbor (KNN) is a well known prediction method that doesn't need any known model in advance, and thus is suitable for the situations when the model doesn't exist or is unknown. However the accuracy of KNN hinders his wider usage in prediction.

Many research found that even though KNN is simple and easy to use, its accuracy is often lacking as compared to the level of model-based methods. KNN calculates the predicted value by taking the average of the K nearest neighbors(usually just a simple mean: divide K into the total values). Consider the situation when some nearest neighbors are far away, the prediction in this case might not be very accurate. For example, if K is 2 and one of them is far away, their average might not be an accurate calculation of the node value under prediction. To improve the accuracy, we propose an innovative approach of Radius of Neighbors (RN) where all nearest neighbors within a certain radius will be taken into consideration for the average.

As an innovative method, RN does not have any related work published so far, though there were/are many researchers working with the widely known KNN [1]. P. Hart proposed the Condensed Nearest Neighbors Rule [2], which reduces a training set in a way that makes a prediction almost 100% accurate. T. Cover devoted to the KNN method in his work “Nearest Neighbors pattern classification” [3]. George Terrel and David Scott were working with KNN as a special case of a variable-bandwidth Kernel density “balloon” estimator [4]. A survey in [5] summarized all the work on using KNN for classification. Please note that our work is different because we use KNN and RN to predict actual values, rather than simple classes to which the predicted value belongs.

After our invention of RN method and its implementation algorithm, we compare KNN and RN methods with respect to accuracy and stability by estimating prediction errors for both. The prediction is an actual numerical value in the range of 0% to 100%, not a simple category value. Our data includes a Training Set of 100 sentences and a testing set of 9 sentences in English. These sentences were translated into three foreign languages Yoruba, Russian and Telugu by their native speakers, who fluently speak both, one of these languages and English. Table 1 shows some sample data we collected for the prediction with an English column, the translated sentences, and incorrect parts in the translation. It also include other factors that we use in prediction, such as search results and frequency grouping of words or sentences.

TABLE I. SAMPLE DATA COLLECTED



#	Sentence	Incorrect part	Correct translation	Google search, results	Frequency group
1	Come here!	.	Иди сюда!	753,000	3.
2	Move away now!	Move away.	Отойди сейчас же!	421,000	2.
3	He is trying to speak.	Speak.	Он пытается говорить.	15,600,000	6.
	...	...	...	...	...
1	Come here!	Come.	ఇక్కడికి రండి	13,700	4.
2	Move away now!	Move.	ఇప్పుడు దూరంగా వెళ్ళిపో	11,700	4.
3	He is trying to speak.	is trying.	అతను మాట్లాడటానికి ప్రయత్నిస్తున్నాడు	4,220	2.
	...	...	...	...	...
1	Come here!	.	Wa sibi	397,000	6.
2	Move away now!	away.	Kuro nbi nsin	839,000	6.
3	He is trying to speak.	Speak.	O fe soro	683,600	6.

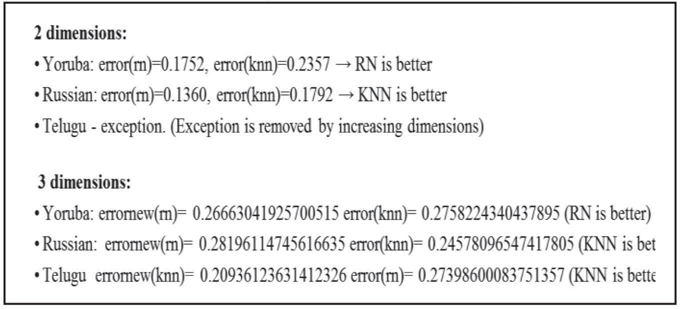
We used “Google Translate” for the translation and manually evaluated the correctness of the translation in each language group with a numerical value. The length of each sentence and the length of the part of each sentence that was translated incorrectly were both found and used to calculate a percentage of translation correctness, which we are using as our prediction result. Another value that we predict is the incorrectness of the translation.

As to the factors to be considered that affect the translation correctness, we are also using a frequency of sentence, as well as frequency of words. This attribute factor was evaluated by copying each sentence into Google search and recording the number of Google search results gotten. For the sake of having a uniform distribution, we divided these outcomes into 6 approximately equal groups. For example, for Russian data set these groups include the following: less than 250000, 250001-500000, 500001-1000000, 1000001-4000000, 4000001-15000001 search results. For our testing set of 9 sentences, the values of the original sentence length and the usage frequency of the translated sentences are known, the correctness or incorrectness is unknown – we estimate it using both KNN and RN. We use the two factors to predict the correctness.

We implemented the above-mentioned KNN and RN methods in a Java program using the following algorithm so that the entire process is automated for us to conveniently extend this work to the translation of other natural languages. The algorithm includes 5 steps:

1) Obtain the list of input sentences to be predicted for translation correctness and create a template to hold each sentence's characteristics.
2) Calculate *incorrectness* of each sentence (incorrect part is divided by the total length)
3) Calculate the *distance* from the sentence to the training set and sort sentences in distance ascending order.
4) Estimate *correctness* using KNN and RN.
5) Calculate standard error of the estimations.

The running of the implementation on three languages gave us the following results (Figure 1) with standard errors for the situations of using 2 and 3 factors in the prediction.



2 dimensions:

- Yoruba: error(rn)=0.1752, error(knn)=0.2357 → RN is better
- Russian: error(rn)=0.1360, error(knn)=0.1792 → KNN is better
- Telugu - exception. (Exception is removed by increasing dimensions)

3 dimensions:

- Yoruba: errornew(rn)= 0.26663041925700515 error(knn)= 0.2758224340437895 (RN is better)
- Russian: errornew(rn)= 0.28196114745616635 error(knn)= 0.24578096547417805 (KNN is bet
- Telugu errornew(knn)= 0.20936123631412326 error(rn)= 0.27398600083751357 (KNN is bette

The above data indicates the following results as given in Table II. Yoruba training/testing data sets gave us better prediction results for RN than for KNN (error of 0.175 vs 0.253). Russian has the other way around and Telugu is inconclusive, which needs future study with more samples.

TABLE II. INITIAL RESULTS

Language	Better Method	
	2 dimension	3 dimension
Yoruba	RN	RN
Russian	KNN	KNN
Telugu	inconclusive as we have mixed results	KNN

Table II summarizes the initial result, which shows that the RN method is promising and has potential of being a better method with higher accuracy in modeless prediction. To further investigate this claim, we identified several research topics that deserve further investigation:

1) Lacking of data within the radius. This problem could be caused by the sampling size being not sufficient. For example, Figure 2 shows the data collected for Telugu with one radius not having any neighbors, and thus undecidable for comparison using this language. We are looking to collect more data with our next version having a training set of millions of sentences provided by existing translation sites.

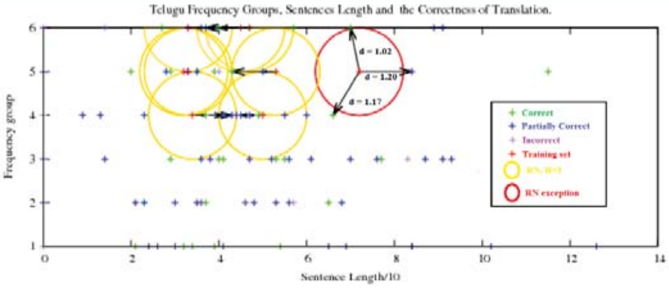


Figure 2. The red radius has no any nearest neighbor

2) Increasing dimension number. The number of factor/characteristics/dimension we used in our initial experiment is only 2 initially. We then increase it to 3 as shown in Figure 3 with an additional dimension of English word frequency. More other factors/dimensions or their combinatorial might affect translation correctness and they are under study.

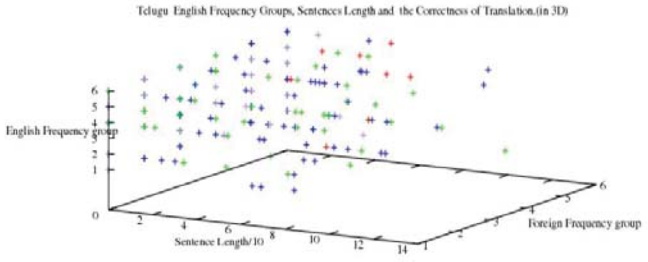


Figure 3. Data with one additional dimension

3) Application of the technology. With the implementation of the KNN and RN algorithms, we were able to predict the correctness of "Google Translate". To make this work useful, we are creating an application for mobile phones users to chat in different natural languages and to be aware of the correctness of the chatting content translation for smooth conversation and improved communciation.

REFERENCES

[1] Nigsch F, Bender A, van Buuren B, Tissen J, Nigsch E, Mitchell JB (2006). "Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization". Journal of Chemical Information and Modeling 46 (6): 2412–2422.

[2] P. E. Hart, The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory 18 (1968) 515–516.

[3] Cover TM, Hart PE (1967). "Nearest neighbor pattern classification". IEEE Transactions on Information Theory 13 (1): 21–27.

[4] D. G. Terrell; D. W. Scott (1992). "Variable kernel density estimation". Annals of Statistics 20 (3): 1236–1265.

[5] L. Jiang, Z. Cai, D. Wang, S. Jiang, "Survey of Improving K-Nearest-Neighbor for Classification", FSKD (1), 2007.