Natural Language Translator Correctness Prediction

Jenny Li J* and Yulia Rossikova, Patricia Morreale

Computer Science, Kean University, 1000 Morris Ave, Union NJ 08340 USA

Received: September 24, 2016; Accepted: September 27, 2016; Published: October 20, 2016

*Corresponding author: Jenny Li J, Computer Science, Kean University, 1000 Morris Ave, Union NJ 08340 USA, E-mail: juli@kean.edu

Abstract

This paper presents a new approach to predictive data analytics, called Radius of Neighbors (RN), and its mobile application, a multilingual RN-Chatter, devoted to improving communication among people, speaking different languages. RN is a modelless method of supervised machine learning, what makes it a fairly simple but effective way of analyzing big amounts of data while keeping an acceptable speed of execution and taking up little run-time space. In the first preparatory stage of our research, we discovered that RN gives better results than well-known K-Nearest Neighbors (KNN) in some cases. We then extended our research to simulating the adjustments of floating radiuses, various volumes of the training data sets and ups and downs of the dimensions to tune RN for its optimum accuracy. We took an experimental approach of not only extending the number of dimensions but, instead, shrinking and modifying them in order to keep the predicted value's neighbors close by within the radius.

Keywords: Machine Learning; Predictive Analytics; Translation Accuracy; Yandex; Google; Live Translation; Multilingual Chatter;

Introduction

Many people around the world use automatic natural language translators, such as Google Translate and its application, on a regular basis. Many of those noticed how sometimes inaccurate and disappointing the actual translation is. At the same time, the translation results significantly vary depending on the languages used. In our previous paper, devoted to the discovery of our new method of machine learning, RN [1], we focused on English-to-Russian, English-to-Telugu, and English-to-Yoruba translation accuracy and closely worked with the people, fluently speaking both to validate our results. We achieved promising results from the initial investigation of this new RN method.

Subsequently, we conducted an online survey, asking people from different cultural backgrounds about their experience with the Google Translate. Results show that there is obviously a lower translation accuracy for languages, spoken in the countries having their major search engines other than Google and providing their own translation services to their users. Among those: Yandex with 62% of market share in Russia and the big influence in surrounding areas, Naver with 70% of market share in South Korea, Yahoo!Japan and Yahoo!Taiwan, widely used in

corresponding regions and Baidu with 63% of market share in China. This one of our conducted survey with its results made us think wider into this topic of language translation.

Rather than analyzing only Google Translate and focusing only on English, we broaden our scope to other languages. Connecting with our survey respondents also helped us obtain initial data samples for language pairs not only involving English, such as German-to-Russian and Italian-to-French. Some of the survey respondents agreed to expertly verify samples of such translation accuracy, which helped us greatly to launch our mobile RN-Chatter application with a big variety of supported languages.

Our main contribution to the topic of intelligent data mining is the innovative idea of applying fairly simple method RN to the complicated topic of translation correction predictions. Through integrating the value of predicted translation correctness in the best way to the chatting application, the messenger facilitates the communication among users using various natural languages. Showing our users on receiving part of the conversation the value of the translation correctness makes them being aware of how trustworthy the provided translation is in real time, which helps to improve the ultimate correctness over time by applying our new method RN method.

This approach of integrating RN to real-time messenger greatly decreases the potential level of overall misunderstanding in international communications. It is a direct path to benefit potentially very large number of users through the friendly interface of RN-chatters. Overall the RN method and its accompanying application, RN-chatter, make it possible to improve international communications, which is now in the mainstream of the Artificial Intelligence and Machines Learning research.

As we see in the news channels on the daily basis, many countries and their people in the world stay completely or partially disconnected from others, losing the ability to be understood and to understand others as the distance grows. This RN-chatter has the potential to connect people and end the era of isolations and conflicts caused by misunderstanding. Many existing messengers do provide language translation features. But as we discussed

previously, their incorrect translation might adversely hinder the smooth communication and understanding. RN translation correctness prediction will help to solve this problem.

Figure 1 shows a typical screen the user sees on an RN-chatter, which looks very similar to conventional messengers such as those provided by Facebook. Because its usage is straightforward with familiar looks and feels, we expect users to start using it rapidly.

The uniqueness of our RN-chatter is the RN value and its multi-linguistic support. As it can be seen from Figure 1, there is a value on the left of the blue phrase received by the user. Our application automatically translated the phrase from the sender language to the receiver language – which in this case is German as shown in blue. The value of 0.000 is the error rate of the translation correctness, which in this case means the phrase is expected to be 100% correctly translated for the user. Currently, the incorrectness of the translation is shown to the users, i.e. the larger number means a worse translation. We have two reasons why using incorrectness rather than correctness. One is to catch the attention of the users and the other is because the current Google translate is quite inaccurate. The main factor, affecting the result of the incorrectness prediction is the frequency of the words used in the sentence, as will be explained in details later.

The important differentiator of RN over other methods is the usage of actual value for prediction rather than classification into categories. For example, while classifiers can predict if a translation is correct or not, our prediction gives a numerical percentage value for the correctness prediction, e.g. a translation with a 78% of correctness or 22% of incorrectness. For now, we have not decided yet what would be the best way of representing this value to our application users (as it was not introduced in any type of user interface before us) and so the decimal value of translation incorrectness is currently shown to the users on the left of each sentence. In this case, a 78% of translation correctness is displayed as 0.220 for incorrectness to draw the attentions of the users.

Figure 1: Chatting area of RN-Chatter

Prediction is a key feature of machine learning. It mines existing data samples to discover patterns in the data and use these patterns to tell what future data might look like. Such data patterns are usually represented in mathematical models, but with RN this is not the case; our method is modeled to speed up the analysis while maintaining similar accuracy.

The main breakthroughs of our method and its accompanying RN-Chatter application include:

It does not need a complex mathematical model to fairly well predict the result based on large amounts of data, but rather use a simple procedure, as described in [1] and briefly later in this article. Subsequently, it provides good high performance, which makes it possible to support thousands of users simultaneously as our performance tests showed.

It is implemented in Java language, run on Apache Tomcat and uses our own sorting mechanism that we called a ShortSort to sort the etalon sentences, used for prediction by distance in 2D, 3D, and higher dimensionally modified spaces.

It does not classify the translation correctness as “good or bad”, but gives an exact percentage of how trustworthy the translation is instead of categories of goodness.

We claim and have proved experimentally while running our tests, that in many cases our method is more accurate than similar approaches (such as for example, KNN), mostly in situations when the data is sparse and the samples are not necessarily clustered around the point of interest. RN takes into consideration all nearest neighbors within a certain radius for the calculation of the average, rather than K number of samples regardless of their distance to it. In this way, RN disposes unuseful farthest neighbors to prevent the prediction results from underserved biases.

Our mobile application performs well while connecting people, physically located in different parts of the world, crossing the search filtering and censoring barriers as its server are located in the US, have good speed and security features. All actions, performed by the application, take no longer than a second, even under possible delays caused by connecting to the search engine, data sorting, and calculation.

Under normal network connection with the absence of malware, the service response delay is negligible that it cannot be captured by the human eye. Figure 2 represents a screenshot of our performance analysis, done for the chatting page under a normal working load using the tool Firebug. As it can be seen from Figure 2, neither process takes more than 37 milliseconds.

Related Work

As an innovative method, RN and its application to language translation correctness prediction are unique and do not have any related work that is already published, though some researchers such as reported by L. Jiang, Z. Cai, D. Wang, S. Jiang in [2] and T. Cover [3] were trying to mitigate similar problems for the purpose of data classification.

There were also researchers working on the improvement of KNN, for example, P. Hart, who proposed the Condensed Nearest Neighbors Rule [4] to effectively reduce a training data set. The problem with such a method is the ability to manipulate training sets, which is lacking in most situations. Our application RN-Chatter uses only real data (its data bank was initially populated with expertly estimated etalon sentences, of which the total amount can be replenished any time, expanded using live chat sentences and phrases and their real-time translation. Such information is provided by search engines directly, as well as the number of search results “on the fly”, which makes it almost impossible to preprocess the training data set using Hart’s Rule; mostly if there are already too many etalon values for many various languages, their approach would not work well.

George Terrel and David Scott were working with KNN as a special case of a variable-bandwidth Kernel density “balloon” estimator [5]. We suppose that our method, RN, may potentially be applicable to such a similar study. In terms of the efficiency of RN and the processing power required by it, Y. Fang, Y. Gao and C.Stap [6] proposed a high-performance infrastructure usable to the RN server to handle a large number of simultaneous users.

One of the most recent studies, related to the topic, is being conducted in South Korea by Taeho Jo [7]. It has some similar ideas of modifying KNN into some sort of RN by emphasizing the Radius as we do, even though he is not planning to apply his research to the prediction of translation correctness in any way or any applications.

Analysis of the above and other related papers leads us to conclude that RN along with KNN can be applied in many applications of the field of intelligent data mining. Those applications include but not limited to stock exchange data, weather prediction services, insurance risks studies, and etc.

Methodology

Our investigation of new RN method included two stages. At the first stage, we implemented both RN and KNN methods in Java and compared them with respect to accuracy and stability by estimating prediction errors for both. In the second stage, we studied the factors that might impact the accuracy of RN so that we can fine tune it into its best performance. This stage

completion allowed us to implement our idea in a RN-Chatter and test them in practice.

We use language translators as our object under study; however, the same idea is applicable to many other data samples. The prediction, in any case, is an actual numerical value in the range of 0% to 100%, not a simple category value of classifiers. As we expected and proved during our research: the closer the given data set are to a uniform distribution, the better our method RN performs in prediction.

RN vs. KNN Study

In the first stage of the research, our data included a Training Set of 100 sentences and a Testing set of 10 sentences in English (which translation correctness was already estimated by our bilingual experts but was not used till the errors of estimate were about to be calculated at the end of the procedure). These sentences were translated into three foreign languages Yoruba, Russian and Telugu by Google Translate and the accuracy of translation was evaluated by the native speakers of those languages, who fluently speak both: one of those and English.

We implemented both methods in order to calculate and compare their accuracy. The entire process is automated. High-level view procedure includes 6 main steps:

- 1) Obtain the list of input sentences to be used for translation correctness and create a template to hold each sentence's characteristics.
- 2) Calculate incorrectness of each sentence by dividing expertly estimated incorrect part of each sentence by its total length.
- 3) Calculate Euclidean distance from new sentence to each of the training set sentences and sort them by distance in ascending order using sentence length and frequency as X and Y. Add the third dimension of English sentence frequency if needed to compare two- and three-dimensional results.
- 4) Estimate incorrectness of new sentence translation using both KNN and RN.
- 5) Calculate standard errors of the estimate (by the regular formula used in statistical studies) and compare them. And
- 6) Check the errors and draw conclusions about the comparison of the two methods.

The research in the initial stage gives us the results as shown in the following Figure 3, including the three initial languages, Yoruba, Russian, and Telugu, using both 2 and 3-dimensional prediction.

As can be seen from Figure 3, for Yoruba language, RN performs better than KNN, for Russian – we got mixed results; Telugu results are inconclusive as no neighbors were found in the particular radius. Summarizing the initial results, we can conclude that the RN method is promising and has a potential

Figure 2: Performance analysis of RN-Chatter by Firebug

of being a better method with higher accuracy in modelless prediction in some situations.

Tuning RN

In the second stage of the investigation, we studied the tuning of RN method to improve its prediction accuracy. We tried to answer questions such as what the best radius for the RN method would be to avoid encountering exceptions, what sample data size will provide us the best translation correctness, whether increasing, reducing the number of dimensions or modifying them will make a difference, and whether hybrid methods, such as a combination of KNN and RN will be a better choice in some cases. To answer these questions, we ran more experiments, collected more data and used different variations of RN method to analyze it.

In the initial study, we used the fixed value of radius for RN-Chatter, taking radius $R = 1$ and using KNN with $K = 3$ nearest neighbors. As even for our three languages, these values did not work the way we expected (there were no neighbors inside the radius of 1 found for Telugu while applying RN and therefore we were not able to analyze KNN results either – had nothing to compare with). This situation gave us an idea of using a hybrid of those while constructing our application: if RN fails to predict as it might happen with a new method, we can use the KNN and the user will still get a value of correctness, even not actually calculated by RN, that is usually better to provide something than nothing or crash with an error.

We also found out that both values of R and K might vary depending on the languages used as well as on the sample size, available in our data bank for a chosen language. While KNN needs fewer data to be collected for its stable performance as the neighbors can be found from far away reducing accuracy, more data needed to find neighbors for the radius of, for example, 1 for RN for any single testing set sentence. Therefore, we continued looking for an optimal solution.

Identifying the best radius

As easy to see, our method requires the very careful choice of the radius with the high level of data/language dependency as without using KNN as a “pillow” our application could give us wrong results or even, hypothetically, unexpected crashes in the case of not finding any neighbors around.

To study the optimum value of radius, we started with the radius of 0.5 incrementing it by 0.01 until it reached 1.5. The unit of the radius will be explained later as well as the counts of sentence length and usage frequency. Figure 4 shows the actual exception caught while analyzing Telugu translation accuracy by Google Translate. As can be seen, there are no results available for the radius $R \leq 1$ as there were no neighbors found within $R = 1$.

As can be seen from Figure 4, the first neighbor was found within the radius of 1.01, the second neighbor within 1.16 and the third one only for the radius of 1.20. Having this number of neighbors not always enough to get good results at least for our

data set. This problem suggested us its own solution: collecting more data will resolve this problem, but app performance must be carefully monitored.

Figure 5 shows some real (straight lines) and estimated (curves of the same color) data of using RN for the English-to-Russian translation, provided by Google Translate for three testing set sentences with the radius from 0.5 to 1.5.

As can be seen from Figure 5, the actual value for the sentence #97 (expertly estimated at the beginning and used for comparison) was 0.1 while depending on the taken radius its estimated value varies from 0.13 to 0.33 and as bigger radius were taken as actually less accuracy our prediction became. It proves that we are not looking for a bigger radius while analyzing the problem but rather for an optimal one.

The best prediction occurred on the radius interval 0.7 - 0.9. The actual value of the sentence #98 was 0.22 but our prediction

2 dimensions:

- Yoruba: error(mn)=0.1752, error(knn)=0.2357 → RN is better
- Russian: error(mn)=0.1360, error(knn)=0.1792 → RN is better
- Telugu - exception.

3 dimensions:

- Yoruba: error(mn)= 0.26663041925700515
error(knn)= 0.2758224340437895 (RN is better)
- Russian: error(mn)= 0.28196114745616635
error(knn)= 0.24578096547417805 (KNN is better)
- Telugu - exception.

Figure 3: Comparison of Standard errors of estimate

Figure 4: RN exception (0 neighbors for Telugu for $R \leq 1$)

Figure 5: Comparison of predicted correctness results and its actual values of English-to-Russian translation for 3 testing set sentences

varies from 0.03 to the 0.18, getting closer to its actual value at the radius interval of 1.3 - 1.4. In this case, another way around, the smaller radius did not provide us good translation correctness. Prediction for the last sentence #99 was pretty far from the original 0.44. From our point of view, it was caused primarily by the nature of the sentence as it is not common from the semantics point of view. "Thank you for the purchase today", with "purchase today" being translated incorrectly, is rarely used by the Russian-speaking Google users as a written statement as e-commerce is not at the same level in the country as, for example, in the US, and the number of returned search results is not right for its translation. At the same time, all words of the sentence are simple and frequent for Google and such a sentence is supposed to be translated more accurately using RN.

Generally we foresee our chatter to be used by a Customer Service type of application in a real business environment and planning to store and provide our users with already collected and stored perfect translation databank without using external calls to the search engine Application Program Interfaces (APIs), which will help to better translate sentences like those and predict the results of such translation with better accuracy.

The idea of increasing the radius while increasing the number of dimensions attracted our attention initially, but the actual solution for it is yet to be discovered in our future work. Finding the "perfect formula" for radius/dimensions number dependency requires more experiments with many more factors involved while, at the same time, our method must stay as simple as possible due to the real-time speed requirement.

The above idea is represented in Figure 6 and left for a future study in our subsequent research projects.

In our opinion, the idea itself makes sense as introducing new dimensions makes data even more sparse, which gives our neighbors more freedom to be far away from each other in the 3D+ spaces. In this case, it also will not solve the RN exception problem as not having neighbors within the radius if 1 of 2D Euclidean space makes it impossible to find them in 3D or 4D without modifying the dimensions. Therefore when the dimension increases, the radius value should be decreased. Future research will answer the question of their ratio of changes to achieve an optimum accuracy.

In conclusion to the topic related to the choice of radius, the error of estimate for floating radiuses for all 10 testing set sentences is provided for the Russian-to-English case. As it can be seen from Figure 7, the error is dropping down at the radius of 0.8 and then mostly stays low, which means that further increasing the radius will not give us a better accuracy. This observation implies that an optimum radius does exist for achieving the highest accuracy with the smallest radius.

While analyzing the data in Figure 7, it seems that the error reaches a bottom of the error rate at the radius of 1.13-1.16. But its further increment at 1.30-1.34 is unclear and can be due to the issue of too small data size.

Identifying the best sampling size

We decided to increase our training data set that included only 100 samples originally. The example of KNN and RN errors of estimate while applying them to the data sets of the sizes 90, 100... 190 is represented below. As can be seen from Figure 8, our method behaves more stable and represents a curve, smoothly approaching to 0.3, while KNN shows some jumping unexpected behavior as some of its neighbors might affect its accuracy from far away data measurement.

The question of whether the bigger sample size is better or if it has an optimum upper bound still needs further research, but as can be seen from the graph above, a bigger sample size provides more stability and better accuracy in case of RN comparing to KNN. The same result was achieved for other languages under analysis.

Figure 6: The idea of increasing radius while increasing the number of dimensions (factors)

Figure 7: Dependency between the value of Radius and the Error of estimate for initial Russian language testing set

Figure 8: Various amounts of training set sizes (from 90 to 190) and their impact on the errors of estimate for KNN and RN

We are looking for the possibilities to collect more data in our own data bank until we have a training set of millions of sentences provided by existing translation sites from different countries. For now, after testing stage of our application RN-Chatter, it has several thousand samples overall for all language pairs altogether, which is not enough to completely rely only on our own resources yet.

Studying impact of dimensions number and their modifications

The regular manipulations with Euclidean distance were not able to find the neighbors in the case of RN exception where there is no data in the radius. However, our approach of shrinking dimensions worked fairly well under the tests. We decided to study the impact of increasing and reducing the number of dimensions to the accuracy of RN prediction, and conducted more experiments.

(a) Increasing the number of dimensions

Taking widely known in the field of Artificial Intelligence approach of expanding the number of dimensions brought us farther from our goal of finding neighbors inside the chosen radius if there are none for the smaller dimensions number.

The number of factors we used in our initial study was only 2. We then increase it to 3 as shown in Figure 9 with an additional dimension of English sentence frequency. More other factors or their combinatorial can affect translation correctness as there are still some semantics rules and special cases that must be taken into consideration while dealing with foreign languages even with our simple approach. Some of those are currently under study.

The figure below represents both training and testing data sets. Those in green, blue and purple represent the main characteristics of our training set sentences, for which the translation correctness is already known, for the red ones we are about to calculate it using our new method of machine learning RN.

As can be seen from Figure 9, the first of our main factors is the sentence length with the initial hypothesis of better translation results for the smaller sentences and worse for the longer ones, which by itself does not perform well as the straight dependency between this factor and actual correctness is not generally true in practice.

We divide our initial English sentences length by 10 for the purposes of simplicity and discretization. Making the sentence length, that may vary from several letters to potentially a hundred of them or more. An integer is also important for data comparison and a balance between our axes, as our frequency groups are also integers, and strictly from 1 to 6 for every language (we chose this policy based on our expertise). We do not want one of the factors to impact the result too much and try to make our data as uniform as possible. Figure 9 clearly shows at-a-glance that our initial data sets were not uniformly distributed and not

every point in our 3D space potentially appearing among tests, will be able to find neighbors close enough to proceed with RN calculations.

Our second and, in some cases, the third parameter are the number of returned results while calling search engines APIs for either already translated and received by our side sentence or initial and translated both, what can be seen, for example, on Figure 9. The initial hypothesis for these factors is that as bigger the number of returned search results as better the expected quality. Figure 10 represents such groups for the initial English sentences, which is the same for any kind of translation (either it is English-to-Russian, English-to-Telugu or English-to-Yoruba). Our third parameter is language-independent. The decision of choosing such groups expertly, depending on the data, brings a challenge of periodically revising them. For now this process is not automated yet and manually done using several known methods of statistical grouping with unequal group intervals.

We conducted experiments for both 2D and 3D for each of analyzed languages. Generally, the third factor did not impact the result much, reducing the errors of the estimate but not significantly different from what is shown in Figure 3.

(b) Decreasing number of dimensions:

We studied the impact of merging dimensions and invented our own method of finding neighbors in case there were none. We recommend applying this approach to various data sets to validate its efficiency for other kinds of intelligent data mining applications. It works fairly well in our experiments and thus it is an approach worth trying for other data analytics.

Figure 9: Example of training and testing data sets (3D)

Figure 10: Dimension Z – English sentence frequency groups

As can be seen from Figure 11, we are mapping our data into the new dimension and our factors for calculating Euclidean distance, initially represented as X & Y, become X and Y/X instead. Our third dimension Z, represented in Graph 9 was taken out from this algorithm. However, there were many ways one can apply to modify the given dimensions. Such variation as $X*X$, $X*Y$, $X*X*X$ or X/Y might work for particular cases. Our main problem is to find a neighbor inside of the circle if there are not out there and we found out that for our case axis X and Y/X work the best.

By applying the new approach to Telugu data, error-exception problem was finally resolved and neighbors were found even for the smallest considered radius of 0.5. The results of this approach are represented in Figure 12.

The Figure shows that there are already 3 neighbors found for the radius of 0.5 for Telugu sentence while there were none even in the radius of 1 before, for the same data. The changes were incredibly easy to implement as well as it is fairly simple to try other possible modifications if some are needed. The code snippet (avoiding some casting details) is represented below:

```
x2=arr.get(i).getEngLength()/10.0;
...
y2= arr.get(i).getRusFreq()/x2;
...
distances[i][k]=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)); //+ (z2-z1)*(z2-z1));
```

Figure 13 shows that nothing else but actually a division of Y by X is needed to update the code for the purpose of using our new RN approach. Everything else is already automated.

Google vs Yandex Study

During our research, we found out, that the accuracy of Google Translate was not very high for some of the languages we analyzed, mostly of the Russian language. It is understandable as most of the native speakers of Russian do not use Google as their main search engine, but Yandex instead, and, therefore, all amounts of data is also not completely reachable by Google crawlers and filtering is used.

As far as we know Yandex also currently uses intelligence dictionary databases, which are not accessible outside of the country. They represent themselves as big data banks of perfect translation from English to Russian and vice versa, which helps to improve Yandex quality significantly. Many famous Russian linguists cooperate with IT professionals for the only purpose of improving translation correctness performed by Yandex. Among those are Elena B. Kozerenko [8]. These and other factors affect English-To-Russian translation accuracy, what we were able to numerically prove by our tests.

At-a-Glance, the difference in the quality of English to Russian translation can be seen from Figure 14. There is no need to show this picture details, its purpose just to show that there is much greener = "correct" for Yandex comparing to Google and redder

= "incorrect" the other way around. As a result of this study, we decided to start with Yandex API and connect to it first while

Figure 11: Mapping Data into New Dimensions

Figure 12: RN exception resolved using new dimension

```
x2=arr.get(i).getEngLength()/10.0;
...
y2= arr.get(i).getRusFreq()/x2;
...
distances[i][k]=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)); //+ (z2-z1)*(z2-z1));
```

Figure 13: Code snippet of dimension modification

	Google	Yandex	Correct translation
131 Over an archway until I came back.	Она встала под арку, пока я не вернулся.	Она встала под арку, пока я не вернулся.	Yandex
132 She didn't realize her class was passing until her teacher mentioned it.	Она не поняла, пока ее учитель не упомянул о том, что ее класс уже начал.	Она не поняла, пока ее учитель не упомянул о том, что ее класс уже начал.	Yandex
133 They won't allow us to sit until everyone else.	Они не позволят нам сесть, пока все не придет.	Они не позволят нам сесть, пока все не придет.	Yandex
134 You will not pass the exam unless you get a 90 or higher.	Вы не сдадите экзамен, если вы не получите 90 или выше.	Вы не сдадите экзамен, если вы не получите 90 или выше.	Yandex
135 I will not tell you anything unless you tell me what you know first.	Я не скажу вам ничего, если вы не скажете мне, что вы знаете в первую очередь.	Я не скажу вам ничего, если вы не скажете мне, что вы знаете в первую очередь.	Yandex
136 Unless you sit here, you will never know.	Если вы не сидите здесь, вы никогда не узнаете.	Если вы не сидите здесь, вы никогда не узнаете.	Yandex
137 When I came in the room, everyone looked at me.	Когда я вошел в комнату, все посмотрело на меня.	Когда я вошел в комнату, все посмотрело на меня.	Yandex
138 I woke up when my baby was crying.	Я проснулся, когда мой ребенок плакал.	Я проснулся, когда мой ребенок плакал.	Yandex
139 I started looking for a gun station when my light went out.	Я начал искать станцию оружия, когда мой свет погас.	Я начал искать станцию оружия, когда у меня погас свет.	Yandex
140 Someone called you while you were at the meeting.	Кто-то позвонил вам, пока вы были на собрании.	Кто-то позвонил вам, пока вы были на собрании.	Yandex
141 We are while we were working at the University.	Мы работали в то время, пока работали в Университете.	Мы работали, пока работали в университете.	Yandex
142 He was staring looking while I was talking on the phone.	Он смотрел на меня, пока я разговаривал по телефону.	Он смотрел на меня, пока я разговаривал по телефону.	Yandex
143 This is where I came from.	Это там, откуда я пришел.	Это там, откуда я пришел.	Yandex
144 Please tell me where you are going.	Пожалуйста, скажите мне, куда вы собираетесь.	Пожалуйста, скажите мне, куда вы собираетесь.	Yandex
145 I need to know where John hid the present.	Мне нужно знать, где Джон спрятал подарок.	Мне нужно знать, где Джон спрятал подарок.	Yandex
146 We will have a picnic whether it rains or not.	Мы будем пикник, будет ли дождь или нет.	Мы будем пикник, будет ли дождь или нет.	Yandex
147 It is time to decide whether we should take action.	Пришло время решить, должны ли мы предпринять шаги.	Пришло время решить, должны ли мы предпринять шаги.	Yandex
148 You need to decide whether or not you are going.	Вам нужно решить, должны ли вы идти.	Вам нужно решить, должны ли вы идти.	Yandex
149 I wrote a letter with the pen you gave me.	Я написал письмо пером, которое вы дали мне.	Я написал письмо пером, которое вы дали мне.	Yandex
150 This is the song that I made with rice and honey.	Это та, которую я сделал с рисом и медом.	Это та, которую я сделал с рисом и медом.	Yandex
151 He cut my hair with his gold scissors.	Он постриг меня своими золотыми ножницами.	Он постриг меня своими золотыми ножницами.	Yandex
152 I am smiling you with my sincere apology.	Я улыбаюсь вам со своим искренним извинением.	Я улыбаюсь вам со своим искренним извинением.	Yandex
153 He came to the front page with confidence.	Он пришел к первой странице с уверенностью.	Он пришел к первой странице с уверенностью.	Yandex
154 We are with me?	Мы с вами?	Мы с вами?	Yandex
155 I can make jewelry.	Я могу делать украшения.	Я могу делать украшения.	Yandex
156 He can't speak French.	Он не может говорить по-французски.	Он не может говорить по-французски.	Yandex

Figure 14: Google (column 3) vs. Index (column 4) English to Russian Translation quality at-a-Glance

implementing our app. There is also one more reason for it – the Yandex services Yandex.Translate and Yandex.XML are free of charge for limited but suitable for us amount of data while Google, Bing, and other APIs took into consideration, have the very short trial period if any, they are not free. Their fees significantly grow as the data usage grow. Yandex provides up to 30000 calls per day free of charge.

The green exterior color on Figure 14 represents a completely accurate translation, red background – unacceptably wrong one; short red sentence chunks represent errors and inconsistencies, which are obvious to the native speaker.

After comparing the English-to-Russian translation of 200 sentences using Yandex and Google, and calculating the errors of estimate for both, we decided to use Yandex Search Engine for our app for Russian and similar Eastern European languages, supported by it. We are also working on accommodating Chinese search engine Baidu for resolving some encoding issues and providing the better translation for some languages supported by it.

Concluding this stage of our research, we can state that the main RN problem, for now, is its error-exception. Handling this problem is strictly related to the training set distribution and more careful choice of training set data will be needed for a future study. We found several solutions and applied them in practice. Among those are a hybrid of RN and KNN and the shrinking-dimension approach.

Application: RN-Chatter

With the implementation of the KNN and RN algorithms, we were able to predict the correctness of Google Translate using the new method of machine learning RN. To make this work useful, we created an application for mobile and other cross-platform users to chat in different natural languages and to be aware of the correctness of the chatting content translation for smooth conversations and improved communication.

For the purpose of demonstration of how the app works, we created a supporting website that provides more information about RN, its authors, and RN-Chatter features. Figure 15 presents a Demo Page of the site.

The application runs on a web server and thus is platform independent. Users can reach it as long as they have any supported web browser. The application is implemented in Java, deployed on Linux server and run on Apache Tomcat, as shown in Figure 15. The data flow diagram is represented in Figure 16 as an architectural diagram.

RN-chatter can be considered as a mobile application, written at Kean University for the general public (universal use). The strongest part of the application – its support of many different natural languages including but not limited to English, Spanish, German, French, Greek, Polish, Russian, Italian and others. Users can register and login into the application and choose the person they want to talk to currently registered users. They do not need to know which language their counterpart speaks (in

case they do not know) as all messages, both sent and received will be translated into the native languages for both sides of the conversation.

As it can be seen in the architectural diagram, we implemented the chatter in such a way that additional natural languages can be added and supported without much change to the entire infrastructure.

As well-known search engine translators currently do not guarantee a 100% correct translation from one to the other natural languages (the expected accuracy varies), we are using our method RN to predict translation accuracy and also display

Figure 15: Screenshot of RN-Chatter website Demo page from RN-Chatter User's manual

Figure 16: Application Data Flow diagram

it to our users. This way they can not only communicate, but understand each other better as a smaller value of translation correctness will prompt them to ask for rephrasing and repeating the message to avoid potential misunderstanding. It will make conversations among people with different backgrounds and cultures much smoother with better understanding.

As can be seen from Figure 17, users from different kinds of Smartphone were testing our application from the different part of the world without any difficulty. RN-Chatter is also accessible from tablets and any other similar smart devices. Since the main service is provided through the server, the end points are platform independent.

The architecture of our application software can be presented using three components as can be seen from Figure 18. We tried to use principles of MVC as much as we can. The model is our database server as shown in the next figure. The controller is the entrance to our servlet Java program which displays the views based on the control analysis.

Besides the data models, the database is also used to log the conversations for improving future translations. The messenger saves all chatting history for the case of unexpected disconnections or future references and allows all previously sent and received messages to be available at any time.

The RN-chatter application website also allows an extra feature for large paragraphs of text translations, i.e., having RN algorithm predicting the quality of normal text translation results. The accuracy is being provided for each sentence in long paragraphs. This feature is not the main one for the RN-chatter, but it would be useful for text editors, a future application under our ongoing studies. The records inside the database to store information of each live chat are shown in Figure 19 of an SQL snapshot.

Figure 19 shows that the chat data for every user and every language is stored in the same data bank, what became possible by resolving some Unicode decoding issues. Hypothetically if RN-Chatter becomes popular tomorrow, the database index implementation can become a bottleneck for successful live operation. Another possible problem can come from network connectivity including potential security risks.

Currently, our prototype is already beyond the firewall and therefore can become a victim of DDOS attacks or another unfriendly usage. The chat data is not yet encrypted that provides a better speed but less protection. Overall performances, including sessions and data pool by now, are under Tomcat configuration.

Hypothetically if RN-Chatter becomes very popular among users, the database calls, related to the search for etalon values for the chosen languages and sorting of the nearest sentences, will provide us a new line of research in the area of the algorithm improvement, concentrating mostly on its efficiency rather than only on accuracy. We are looking for ways to improve both efficiency and accuracy simultaneously.

If a lot of data on many languages becomes available, we may need to rethink our sessions and database-connection

implementation, or create more indices in the database, such as a (sourceLanguage, targetLanguage) pair or potentially use a Cloud infrastructure to make the application more scalable.

In our opinion, RN-Chatter can be considered as a very unique and perspective product. It can also be quickly expanded, as it is implemented in Java – one of the most popular object-oriented languages with state-of-art multithreading, which can potentially accommodate millions of RN-Chatter users. When the number of users gets really large, over the limit of the thread number, we can use a grid of processors such as Graphic Processing Unit

Figure 17: RN-Chatter on iOS and Android

Figure 18: Architecture diagram

translate_id	postname	sourcelang	source	targetlang	translation	stamp	receiver	nsource	grou	notranslate	quality
1...	B	de	Guten nacht	en	Good night	2015...	A	158253...	6	158253...	6 0.450
1...	A	en	How are you?	de	Wie sind Sie?	2015...	B	746470...	1	110447...	6 1.000
1...	B	de	Gut	en	Well	2015...	A	941398...	1	941398...	1 0.833
1...	A	en	ok just deploy	de	ok nur noch b...	2015...	B	6501934	5	240783	1 0.080
1...	B	de	ja	en	Yes	2015...	A	789227...	6	789227...	6 0.000
1...	A	en	Hello B	de	Hallo B	2015...	B	244057...	1	4934331	6 1.000
1...	B	de	Hi	en	Hi	2015...	A	462105...	6	462105...	6 0.000
1...	B	de	Hallo A	en	Hello A	2015...	A	414093...	1	414093...	1 0.833
1...	A	en	How are you?	de	Wie sind Sie?	2015...	B	796970...	1	100497...	6 1.000
1...	B	de	Ich sind gut	en	I are well	2015...	A	105102...	1	105102...	6 0.450
1...	A	en	It is good	de	Es ist gut	2015...	B	922466...	1	104073...	6 1.000

Figure 19: MySQL snapshot of the Translate Table

(GPU) to automatically accommodate the increase of usage loads. Examples of this line of work are given in [9, 10]. We will use such technologies in our production environment of the RN-chatter server. Some of the Java classes can be seen in Figure 20.

Currently RN-Chatter directly supports the following languages: English, German, Albanian, Armenian, Azeri, Belarusian, Bulgarian, Catalan, Croatian, Czech, Danish, Dutch, Estonian, Finnish, French, Greek, Hungarian, Italian, Latvian, Lithuanian, Macedonian, Norwegian, Polish, Portuguese, Romanian, Russian, Spanish, Serbian, Slovak, Slovenian, Swedish, Turkish, and Ukrainian. The language choice menu is represented by the drop down list on the application. More generally – these are mostly European languages, but not of Asian ones, what resulted from some decoding difficulty, solving which will bring to the application many other languages such as Arabian, Bosnian, Chinese, Georgian, Hebrew, Icelandic, Indonesian, Japanese, Korean, Malay, Maltese, Thai, and Vietnamese, all of which are supported by Yandex Search and Translate.

As the number of supported languages grows, the performance of the application becomes crucial. As can be seen from Figure 22 there is a limited number of steps that make the whole procedure work and none of them can be skipped, for now, therefore, every action we perform must be implemented optimally making our application work efficiently.

Results And Conclusions

This paper presented an innovative RN method for modelless data prediction, an improvement over traditional machine learning KNN method. Our research so far identified the situations where RN is best at, in terms of prediction accuracy, for natural language translations. We have also attempted to fine tune RN parameters to reach its best performance in terms of accuracy and performance. The investigation of RN method and its application is ongoing with a list of additional research topics presented at the end of the previous sections.

As a new method of machine learning, RN is unique and

Figure 20: UML diagram

Figure 21: RN-Chatter Language choice drop-down

Figure 22: Message transmission, search and RN calculation sequence diagram

significant, as it might provide better predictions than other known modelless methods and can be applied to the fields as broad as artificial intelligence, machine learning, intelligent data mining, big data, mobile applications [11] and cyber security [12].

Besides the theoretical advancement, we applied the new RN method to the creation of an RN-chatter mobile application that supports instant communication among people of various natural languages. The design and implementation details of the application, as well as the accuracy of the language translations are presented in the paper.

We hope our work will contribute to the fields of artificial intelligence, machine learning and intelligent data mining, and to attract more researchers to participate in the research of this promising topic.

Acknowledgements

This research was supported through Kean University SpF 2015 and 2016 Summer Research Program. Its presentation on Int'l Conf. on Artificial Intelligence ICAI'15 [1] and IEEE Int'l Conf. on Intelligent Data and Security IDS'2016 [13] were also sponsored by the college. We would also like to thank Council on Undergraduate Research (CUR) for selecting this work for

presentation at Posters on the Hill 2016. The support of the Kean University Foundation and ORSP is greatly appreciated.

References

1. Y. Rossikova, J. Jenny Li, P. Morreale, "Predicting Correctness of Google Translate" proceedings of the International Conference on Artificial Intelligence ICAI_2015, held at WORLDCOMP 2015 Congress, pp.825-826, Las Vegas, July 27-30, 2015. ISBN:1-60132-405-7-X, 2015 CSREA Press, USA.
2. Jiang L. Cai Z, Wang D, Jiang S. Survey of Improving K-Nearest-Neighbor for Classification. FSKD. 2007.
3. Cover TM, Hart PE. Nearest neighbor pattern classification. IEEE Transactions on Information Theory. 1967;13(1):21-27.
4. Hart PE. The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory. 1968;14(3):515-516.
5. Terrell DG, Scott DW. Variable kernel density estimation. Annals of Statistics 1992;20(3):1236-1265.
6. Park JJ, Zomaya A, Jeong HY, Obaidat M. Frontier and Innovation in Future Computing and Communications. Springer Netherlands. 2014:127-134.
7. Taeho Jo. KNN based Word Categorization considering Feature Similarity. Proceedings of the International Conference on Artificial Intelligence. 2015:343-346.
8. Elena B. Kozerenko, "Linguistic Constraints for Statistical Machine Translation", Proceedings of the 2010 International Conference on Artificial Intelligence, ICAI 2010, July 12-15, 2010, Las Vegas Nevada, USA, ISBN 1-60132-148-1, pp. 927-933 2015 CSREA. Press, USA.
9. Gao Y, Iqbal S, Zhang P, Qiu M. Performance and Power Analysis of High-Density Multi-GPGPU Architectures: A Preliminary Case Study. In the proc. of 17th IEEE International Conference on High-Performance Computing and Communications 2015: 66-71. doi: 10.1109/HPCC-CSS-ICESS.2015.68
10. Zhang P, Gao Y. Matrix Multiplication on High-Density Multi-GPU Architectures: Theoretical and Experimental Investigations. Lecture Notes in Computer Science. 2015;9137:17-30.
11. Zhang P, Gao Y, Fierson J, Deng Y. Eigenanalysis-based task mapping on parallel computers with cellular networks. Mathematics of Computation. 2013;83(288):1727-1756.
12. Bassu D, Izmailov R, McIntosh A, Ness L, Shallcross D. Centralized Multi-Scale Singular Vector Decomposition for Feature Construction in LIDAR Image Classification Problems. Applied Imagery Pattern Recognition Workshop 2012.
13. Rossikova Y, Jenny Li J, Morreale P. Intelligent Data Mining for Translator Correctness Prediction. IEEE Intelligent Data and Security. 2016.