Bilingual Translation

Prajval M[1], Ishaan R.D[2], Tejaskumar K[3], Spoorthy V[4], Shashidhar G Koolagudi[5]
Department of Computer Science and Engineering
National Institute of Technology, Karnataka
Email: [1]26prajval98@gmail.com, [2]ishaanrd6@gmail.com, [3]tejasnitk@gmail.com,
[4]vspoorthy036@gmail.com, [5]koolagudi@nitk.edu.in

Abstract— **Nowadays, many times when people have a conversation in their mother tongue, they tend to include words from another language in the conversation just because the subject being referred to by that word is easier to tell in that language. In this paper, an approach is proposed to identify English words being used in Kannada sentences, and then translate those Kannada sentences with an English word into grammatically and semantically correct complete Kannada sentences without changing the context of the English word. The dataset is scraped from Wikipedia and a dictionary of English words to Kannada from Shabdkosh. An N-gram model is used to predict the likelihood of each possible sentence with all translations of the English word and predict the best sentence. The scope of the proposed model is for bilingual data. This may be useful in improving the speech recognition abilities of Artificial Intelligence (AI) assistants that are now found almost anywhere. The N-gram models used provides an efficient solution for the Bilingual Translation problem.**

Index Terms— **N-Gram model, bigram models, trigram models, Artificial Intelligence, NLP, internet scraping**

I. INTRODUCTION

With the advent of computers, translation from one language to another has helped people from different parts of the world to have conversations easily. New technological advancements in the field of speech recognition over the past decade have made the translation of languages more seamless. Several technologies have been developed for machine translation over the years.

Natural Language Processing (NLP) is a field of Artificial Intelligence that allows computers to understand and comprehend human spoken languages. Usually, computers understand only structured data like C, C++ which is defined using a set of rules but when it comes to unstructured data there are no concrete rules which govern the language, so we need a different method in order to make the computer learn. Over the years, Machine learning (ML) has been an important and powerful method in this dimension with Deep Learning used to perform text summarization, named-entity tagging, etc. It has totally rewritten our approach to machine translation. Machine learning researchers who know almost nothing about the languages have provided simple machine learning solutions that are beating the best expert-built machine translation systems in the world.

One of the important problems in machine translation is to preserve the context of the sentence when the sentence in the old language is translated into the new language. Various sentence evaluation schemes are used for this reason. In this paper, we are converting a Kannada sentence with frequently used English words to a Kannada sentence with no English words preserving the context. Many NLP models can be developed for doing this task. N-gram models are used for this task.

Major contributions of this paper include:

- Scraping of the internet for data
- Creating and using of N-gram model for translation of the different language words in a sentence
- Usage of previously trained N-gram words in this mechanism
- Efficient and fast training with small language datasets

In Section 2, we discuss related work. The Dataset Description is provided in Section 3. The proposed approach is presented in Section 4. In Section 5, the experimental results and analysis are presented. The conclusion is presented in Section 6. In Section 7, the further scope of the project is presented.

II. LITERATURE REVIEW

This section is about the literature review performed for this research process

A. *MACHINE TRANSLATION*

Machine translation is a part of NLP which is used to translate a given sentence from a source language to a different target language. It is a process that inputs a bilingual dataset to build language models in order to translate the text from the source language to the target language. Some of the approaches include: Rule-Based MT, Direct Based MT, Corpus-Based MT, and Knowledge-Based MT [1].

1) RULE-BASED: RBMT system parses the source text into an intermediate representation based on certain grammatical, lexical and morphological rules. The intermediate representation is then converted to the target languages. There are two schemes to RBMT: Transfer Based MT and Interlingual MT.

2) Direct Based: Direct translation, also called a word by word translation, translates an individual word in a sentence from one language to another using a dictionary.

3) Corpus-Based: CBMT depends on the analysis of bilingual text corpora. Statistical MT (SMT) and Example-Based MT come under the Corpus-Based MT. SMT is good for catching exceptions to rules when translating the English source sentence to the target sentence. The benefit of SMT is linguistic knowledge is not required for translation. The creation of a parallel corpus is one of the troublesome steps in the SMT system.

4) Knowledge Based: A domain-specific Knowledge Base (KB) is used in KBMT for translation. This takes semantics and ambiguity into account. The KB has to be created based on ontology and semantic web.

5) Hybrid Based: HBMT is a combination of two or more MT techniques. Failure from a single technique is overcome using this method. The techniques used to run in a parallel manner and the output is the combination of all the outputs of the techniques used.

B. N-GRAM MODEL

N-gram model[5] is a method that gives the probability of a target word given n context words. This model is used for spelling mistake detection systems, sentence evaluation systems, etc.

So in n-gram models, The probability of the entire sequence of words with a given history is calculated. The sequence of N words is represented as $w_1........w_n$ or w_1^n. The entire sequence probability can be written as:

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2).....P(w_n|w_1^n-1) \\ &= \prod_{k=1}^{n} P(w_k|w_1^k-1) \end{aligned} \tag{1}$$

C. BIGRAM MODEL

This is a type of N-gram model where the number of context words used to evaluate a sentence is 1. The search window will thus be of size 2 and hence the bi-gram model.

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-1}) \tag{2}$$

D. TRIGRAM MODEL

This is a type of N-gram model where the number of context words used to evaluate a sentence is 2. The search window will thus be of size 3 and hence the tri-gram model[3].

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-2}w_{k-1}) \tag{3}$$

E. PERPLEXITY

Given a sentence, the perplexity of a sentence measures how unlikely the occurrence of that sentence is. The lesser the value of perplexity more likely the sentence is likely to occur. A model can be evaluated by the measure of the perplexity of the model. Lower the perplexity better the model is.

We denote the perplexity as PP, sentence as s and number of words in a sentence as n throughout the paper.

The evaluation of sentences is done using n-gram models. These help in getting the perplexity of the sentence. The perplexity of a sentence is defined as follows:

$$PP(s) = \sqrt[n]{\prod_{k=1}^{n} \frac{1}{P(w_k|w_1...w_{k-1})}} \tag{4}$$

So for bigram model the perplexity formula will be as follows:

$$PP(s) = \sqrt[n]{\prod_{k=1}^{n} \frac{1}{P(w_k|w_{k-1})}} \tag{5}$$

and for trigram models the perplexity will be as follows:

$$PP(s) = \sqrt[n]{\prod_{k=1}^{n} \frac{1}{P(w_k|w_{k-2}w_{k-1})}} \tag{6}$$

III. DATASET DESCRIPTION

The dataset required to train our model has been scraped from the web and English-Kannada word translations have also been scraped from the web.

The main dataset consists of grammatically and contextually correct Kannada sentences. The biggest dataset has around 13k Kannada sentences and 112k tokens.

Dataset was compressed using XML and the sentences can again be generated by extracting the sentences from it. The testing dataset has about 1326 Kannada sentences with an English word in it and the possible set of Kannada words that can be used.

IV. PROPOSED APPROACH

The Kannada sentence with an English word is given as input. English word detector detects[4] the English word present in the sentence and searches the web for all the equivalent meanings on the web. Once they have been found, sentences are generated replacing the English word with all the possible Kannada words using English to Kannada dictionary. The changed sentences are given as the input to the Bi/Trigram model as shown in Figure 1 and the most likely sentence is picked based on the least value of perplexity.

The reason for choosing this model instead of RNN is due to unavailability of grammatically and contextually incorrect Kannada sentences. Hence the LSTMs or any other RNNs cannot be trained to detect incorrect sentences.

Bi/Trigram model predicts the occurrence of the next word given the previous one/two words. The block diagram shows the working of the model. Also, comparisons between the two models with different corpus sizes are given.

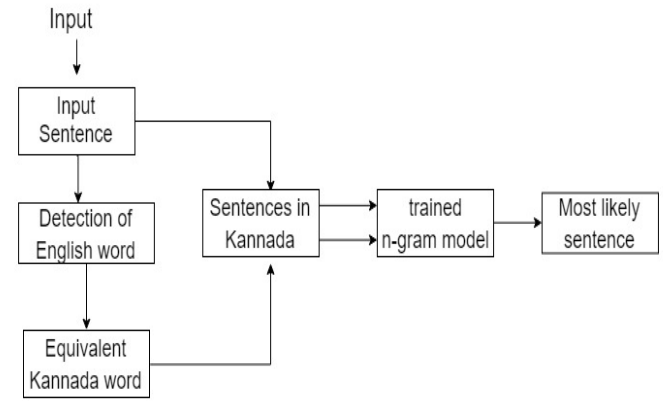

Fig. 1. Input sentence consists of Kannada sentences with at most one English word which is passed on to train the N-gram model. The model predicts the most likely sentence.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The implementation of this model has been done using Python 3.6 and nltk. The dataset (Kannada sentences) were obtained after scraping the Wikipedia pages. The dictionary used is an online dictionary. The model is highly efficient even on a small dataset and very fast to train. The model was trained on a windows system with i7-7700HQ processor, 16GB DDR4 RAM and GTX 1060 6 GB GPU.

The training process is dynamic so the model keeps updating every time it is trained or sees a new sentence. The trained data is stored as a JSON file so that it can be used anywhere by anyone. The sentences are NULL padded left or right depending on the sentence.

As perplexity of a sentence is inverse to its probabilitys nth root where n is the length of the sentence, the probability can be 0. Also whenever a new word, not in the corpus of the training set comes, the word will not be recognized and all the probabilities of the sentences formed will be 0.

All these leads to division by 0 error. Hence they are given very small probabilities. Hence more likely sentence cannot be picked.

The model's accuracy was tested with a test set which had 1326 sentences with each sentence having one English word. The correct substitution of the word is given as an array to which the output word is compared to. This was done on all the test set sentences,

For different corpus size and different models, we get different accuracies.

- For corpus size of 30k tokens, bigrams give an accuracy of 70.6 percent whereas trigrams give an accuracy of 68.9 percent. The bigrams are better when corpus size is less because of the number of context words required to predict the target word is less i.e the second word in the trigram might now occur in the tokens.
- For corpus size of 81k tokens, bigrams give an accuracy of 73.4 percent whereas trigrams give an accuracy of 76.4 percent. The trigrams are better here because of the increase in the number of tokens increases the possibility of the second context word being present in the model, thus almost all the 2-word pair will be present in the model. For all trigrams, the bigrams are a part of it. But the bigrams have lesser accuracy due to a fact that the single context word will have many options and the bigrams might choose a different token.
- For corpus size of 112k tokens, bigrams give an accuracy of 78.5 percent whereas trigrams give an accuracy of 82.3 percent. In this case, all the 3-word pairs have been mostly saturated. The reason why trigrams provide a better accuracy is the same as the above case.

Corpus Size	Bigrams	Trigram
30871	70.6	68.9
81530	73.4	76.4
112132	78.5	82.3

TABLE I

CORPUS SIZE VS ACCURACY PERCENTAGE FOR BIGRAMS AND TRIGRAMS

However, with the further increase in N for N-gram models the accuracy will decrease. This is because of N-gram is a subset of N-1 gram and with an increase in N, the target word might not even appear for a given set of N-1 words. Hence bigrams and trigrams are a sweet spot.

The model's accuracy increases with an increase in corpus size. Bigram and Trigram models are almost equally accurate as shown in Figure 2. However, when the model has been trained with lesser data set the bigrams to provide better accuracy compared to trigrams. Hence, for machine translation purposes Bigram models can be used as they are computationally less intensive.

Presently, there is no implementation of the given problem description, however, Google translate which translates English sentences to Kannada sentences has an accuracy of 60 percent. Also their model uses LSTMs (RNNs) for machine translation purposes. Any machine translation model can be improved by adding this model to their output to evaluate how likely the sentence is good for many sentences of different length putting a stopping condition on length of the output sentence or keeping a threshold probablity.

VI. CONCLUSION

The trigrams produce an accuracy of around 82 percent and bigrams about 79 percentage. Both Bigram and Trigram models produce results similar to each other. Trigrams are a bit better compared to bigrams which give lesser perplexity compared to bigrams and better accuracy. Perplexity is a measure of how bad a sentence is so a model which gives lesser perplexity for the same sentence compared to other models is considered better. For smaller datasets, it can be seen that bigrams are a better model compared to trigrams. The English word present in the Kannada sentence is

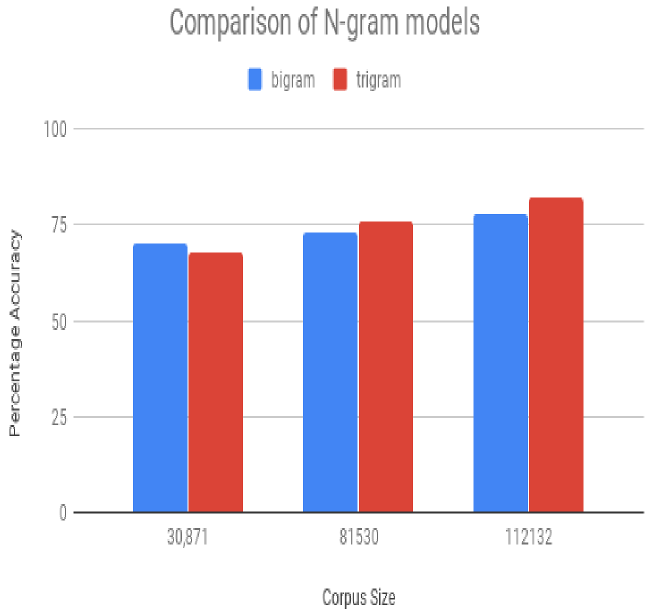

Fig. 2. Comparison of different N-gram models.

converted to give a sentence completely in Kannada. Words in English are first converted to their corresponding Kannada words using a dictionary. English words having multiple meanings when translated to Kannada are segregated based on the probability of occurrence of that particular Kannada meaning of the word in the given the sentence. This is done by the n-gram models which take into account the context of the word in a sentence to predict the most likely sentence formed by the given word.

These models can be used in the selection process of the best sentence in machine translation. The time taken to train these models is significantly very less compared to RNNs also even smaller datasets produce a better accuracy percentage compared to RNNs. These models will be very useful in Machine Translation from or to a language which does not have a very high content on the internet.

VII. Further Scope

The model that has been used for translation of English to Kannada can further be extended for the process of a complete translation of English sentences into Kannada sentences preserving the context of the sentences. This model can also be used to translate from English to any other regional language if a big dataset for training the model is available. Instead of taking text as input for the model speech can be made as input. Bigger N-gram models can be used instead of Bi/Trigram models and accuracy can be compared. Further, this model that is created can be used in real-life applications for the people who speak a mix of Kannada and English language. Such people can make use of this kind of translator to get a proper Kannada sentence which would aid them in speaking.

References

[1] Nair, Jayashree, et al. An Efficient English to Hindi Machine Translation System Using Hybrid Mechanism. 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI), 2016.

[2] Singh, Shashi Pal, et al. Machine Translation Using Deep Learning: An Overview. 2017 International Conference on Computer, Communications and Electronics (Comptelix), 2017.

[3] Brugnara, F., and M. Federico. Techniques for Approximating a Trigram Language Model. Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96

[4] Parameswarappa, S., and V.n. Narayana. Sense Disambiguation of Simple Prepositions in English to Kannada Machine Translation. 2012 International Conference on Data Science Engineering (ICDSE), 2012.

[5] Gao, Jianfeng and Nguyen, Patrick and Li, Xiaolong and Thrasher, Christopher and Li, Mu and Wang, Kuansan. (2010). A Comparative Study of Bing Web N-gram Language Models for Web Search and Natural Language Processing.