International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 09 | Sep 2020 www.irjet.net p-ISSN: 2395-0072

Named Entity Recognition using Word2vec

Kumarjeet Poddar[1], Pramit Mehta[2], Vaishali Gatty[3]

[1]Student, Vivekanand Eduction Society's Institute of Technology, Mumbai
[2]Chief Technology Officer, SETU INDIA, Mumbai
[3]Assistant Professor, Dept. of MCA, Vivekanand Eduction Society's Institute of Technology, Mumbai

---

Abstract - *Food recipe data is very important for preparing new dishes and these data is available in the various online platform but in unstructured form and no relationship between them. To solve this problem, the model is developed for recipe data using the word2vec model. This model trying to extract ingredient names and quantity from phrases or sentences and is also find most similar or identical ingredient names. This is only based on word2vec and natural language processing.*

Key Words: Natural language Processing, Food recipes data, Information retreival,Word2vec.

1. INTRODUCTION

Food recipes are very essential for individuals and millions of data available on the internet which is submitted by different users of different countries. These data contain recipes name, ingredient name, quantities with units and some extra comments but in unstructured form. There are some algorithms which are CRF(Conditional Random Feilds) based which has got success in extracting and structuring recipes data, but it can't find similarities between them. The system which is built on them is also very useful to the developed recommendation system.

So we developed a model with help of word2vec and some rule based tagging to find ingredient name inside phrases or sentences and also similarities between ingredient name with other words inside phrases or sentences. Which helps to determine the relationship between the words.

2. Technology Used

Natural language processing now a days very important for text mining. NLP provides libraries like Nltk and SpaCy which is very useful. Most of, I used Spacy libraries in this model for identification of noun and part of speech tagging. Nltk is also good for auto-tagging. Rule Based approach also gives advantage at many stages of coding.

Web Scrapping very essential for extracting data from websites. For recipes related data I used recipe_Wikia[3] website. Extracting data from websites with the help of web scraping and clean that data Using Rule Based Approach. After this step push into PostgreSQL relational Tables. This is all for preparing Dataset. We are using the Word2vec model which is a neural network based So Dataset Size must be large for a more accurate result.

Word2vec is a technique for natural language processing that uses a neural network model. It requires a large corpus of Text data. This model can detect synonyms word or suggest additional words for words or Phrases. In this model, word is converted into vectors so we can easily perform any vector operation. It uses cosine similarity for detecting similar words. It produces word embedding. In our model after extracting a noun from sentences or Phrases using Rule based approach and Spacy library word2vec try to predict is that noun is an ingredient name or not. If got an ingredient name than trying to predict cosine similarities between other words inside the sentence. In that way, context is going to be determined.

For this word2vec uses two types of architecture. Basically first is CBOW(Continuous Bag of words) and another is Skip-Gram.

In the CBOW architecture, the model is going to predict the current word from a window of surrounding context words. In continuous skip-gram architecture, the model trying to predict context words with uses of the current word in the given window.

Programming language uses python which best suitable for this type of work.

3. Proposed Solution

The other models also extract all recipes details and also converts it into a structured format, but it won't specify the relationship between words precisely the reason why we have used Word2vec model. With the use of Word2vec model context words can predict the target word easily.

With the use of our model ingredient name and quantity can be extracted from the sentence also with the ingredient name we can find out the best context. Other different words with the ingredient name can be predicted with the use of word2vec model.

4. Ingredient Details Extractions

Word2vec in Named Entity Recognition is very useful. Our model used part of speech tagging with word2vec for getting accurate target word.

Test data for model:-
2 tablespoons unflavored gelatin, dissolved in 1/2 cup water

Our model uses training Dataset and produce the result:-
{2 tablespoons, quantity},

{gelatin,ingredient_name},

{½cup,quantity},{water,ingredient_name}

For the Context of target word first need to define window size and depending of window size we will get context words of target word(Ingredient name). It's totally depend on cosine similarity between the words in particular window.

5. Experimental Elaboration

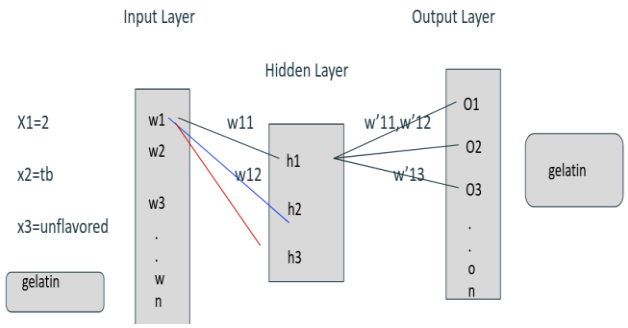
Word2vec algorithm uses neural network so large training data is required. We used PostgreSql for storing data after cleaning. For efficient work of the model we have used approximately 10 thousands recipe details and 3 thousands of ingredient name in relational format. We also use data from kaggle.

Using Spacy library sentence have been tokenized and Noun and Pronoun is extracted from it , after that word2vec checks for the ingredient names. After finding ingredient name model can also predict context(the possibility of number of occurrence of different words in the context). The Best context from the target word is obtained.

5.1 Steps in Technique

Model goes through various steps like forward pass, Error Calculation , Backword pass.

Consider an Example:- 2 tablespoons unflavored gelatin. Here gelatin is ingredient name and it is target word for the model. Model is going to first predict gelatin.



Word2vec model have two types of architecture:- CBOW and SKIP-GRAM which consist of of three layer:- 1) Input Layer 2) Hidden Layer 3) Hidden Layer.

Number of nodes at input layer and Output Layer depending upon the vocabulary size we use I.e training Dataset.

Number of Nodes at Hidden Layer Represent Dimensionality.

Main Task of model to find target word(Ingredient name) into sentences or phrases.After that Skip-Gram Model is going to determine Context of target word.

First we train the system ,for training we give input as sentence or phrase and Apply forward pass, calculate Error, If Error detected perform backword pass.

The Main reason behind the backword propagation is to optimized the weight.

5.1.1 Forward Propagation

We calculate weight at node of Hidden Layer. For calculation:-

$$\begin{bmatrix} h1 \\ h2 \\ h3 \end{bmatrix} \cong W^T \times X$$

It is look like

$$h1 = W_{11}x_1 + W_{w21}x_2 + W_{31}x_3 + W_{41}x_4 + \ldots + W_{1n}x_n$$

Here W11 is weight associated with neurons and X1 is input parameter.

Weight calculation at hidden layer to output layer .In this we perform two operation 1) association 2) Softmax function Association means sum of weight, generally its meanings is sum of weight of all neurons connect to every output Layer and associated hidden layer weights.

$$Net(O) = W'^T \times h_i$$

$$Net(O_1) = W'_{11} \times h_1 + W'_{21} \times h_2 + W'_{31} \times h_3$$

Now Calculate Softmax Output:-

$$Out(O_i) = e^{Net(O_i)}/(\sum_{1}^{n} Net(O_i)$$

more generalized

$$Out(O_1) = e^{Net(O_1)}/(e^{Net(O_1)} + e^{Net(O_2)} + \ldots + e^{Net(O_n)})$$

5.1.2 Error Calculation

Let W_0 is actual output and W_I is is given context words and V is size of input context.
Training objective is maximize the conditional probability of actual Output word given context words thus loss function is

$$max(log(Y_{j*}))$$

Here **Y** represent softmax output.
As we want to minimize the error
E = -log P(W0/WI).
With Than We calculate derivative

$dE/du_{j*} = Y_{j*} - t_j$ Here $t_j = 1$ if $t_j = t_{j*}$ else tj=0when tj represent actual output word.

5.1.3 Back Propagation

Update the weight between Output Layer to Hidden Layer
$(W^1_{11})^{new} = (W^1_{11})_{old} - n\ (dE(O_1)/dW^1_{11})$
where **n** is learning rate.
Update weight between hidden layer to Input layer.

$(W_{11})^{new} = W_{11} - n\ (dE/dW_{11})$

These all Operation performed into recipes data for extracting and finding the context of target word.

5.2 Evaluation

Accuracy of model on training data is approximately 0.88.
Accuracy of model on Testing data is 0.99.

For getting more precise and accurate result need huge corpus of data because word2vec uses neural network.

Some cases when ingredient name may be unit name like e.g clove in that case our training dataset will be useful. Context is helpful for this type of cases for the model.

6. Conclusions

With the use of Word2vec, we have produced a good result for Named Entity Recognition, We can use it in the recommendation system.

The ingredient name is taken as target word with the help of part of speech tagger and Word2vec. A rule-based approach is also followed in the process.Word2vec uses large datsets, the larger the dataset better the accuracy.

With the help Word2vec, the Name Entity Recognition system has been improved to best standards still there is a gap for improvement.

For time being we just have worked on the imperative sentences, there can be future expansion. Word Conflicting only resolve by our training dataset.

REFERENCES

[1] Thomas Mikolov, Ilya Sutskever, Google Inc. "Distributed Representations of words and Phrases and Their Compositionality"

[2] "Information Extraction from Unstructured Recipe Data" By Nuno Silva, David Ribeiro,Liliana Ferreira, Research and development, Fraunhofer Portugal AICOS.
https://dl.acm.org/doi/10.1145/3323933.3324084

[3] https://recipes.fandom.com/wiki/Recipes_Wiki