
Computational Linguistics: Text Prediction and Sentence Correction

✉ ¹ E. Padmalatha, ² S. Sailekya

¹ CSE Dept, CBIT, Gandipet, Hyderabad, Telangana
padmalatha@cbit.ac.in

² Prowess ENTP, Vishakapatnam, Andhra Pradesh
lekya.sheral@gmail.com

Received: 28th May 2020, Accepted: 18th June 2020, Published: 31st August 2020

Abstract

Language in this growing technology world text became the medium to communicate socially. Even though desktop computers have existed since a long time, the method of typing and feeding input has not changed much. Versions after versions of popular text editors have come, and yet no editor has addressed the difficulty of predicting the next possible word and correction of predicted sentence. Also, predictive editors cease to exist in desktop computers even today. This paper aims to predict the next frequent word according to the trained corpus using the n-gram model and even checks the spelling of the entered word. In addition, it checks for the correction of sentence according to grammatical rules. This in proposed method it explores the use of a new software for the input on desktops, which relies on a dynamic predictive algorithm using n-grams and suffix trees to significantly reduce the effort of typing.

Keywords

Natural Language Processing, Spell Check, HMM, n-gram, Corpus.

Introduction

The proper meaning of a sentence could be acquired through a machine learning approach is called natural language processing. Basically it represents computational model of human language processing. Computational linguistics is the scientific and engineering discipline concerned with understanding written and spoken language from a computational perspective, and building artifacts that usefully process and produce language, either in bulk or in a dialogue setting.

Some of the most prominent are: efficient text retrieval on some desired topic; effective machine translation (MT), question answering (QA), ranging from simple factual questions to ones requiring inference and descriptive or discursive answers, text summarization, analysis of texts or spoken language for topic, sentiment, or other psychological attributes.

Text Prediction and Sentence Correction are the essential concepts of linguistic structure and analysis. A statistical and machine learning technique in natural language processing plays an important role dealing with those applications. NLP is the application of the HMM. Probability of occurrences of a sentence is predicted using HMM.

At present the most commonly employed declarative representations of grammatical structure are **context-free grammars** (CFGs) as defined by Noam Chomsky (1956, 1957) [6], because of their simplicity and efficient parsability. Markov's n-gram model deals with the text prediction and Chomsky's Context free grammar deals with sentence Correction [6].

This paper aims at correcting the spellings of word, predicting the next word using n gram [3] model and checks for the correctness in the syntactic structure of the sentence using grammatical rules in a single application. In addition, it aims at improving the efficiency of storing and retrieving the data from large corpus data with reduced time and space complexities using hash map implementation.

Related work

Now-a-days every mobile application has auto correct module which suggests the spellings and the next word. Few Desktop application like Microsoft Word suggest the spelling and detects punctuation errors. With the increasing corpus, predicting the next word has become easy. But the predicted sentence may not be appropriate. Problems in Existing System are there is no such application for desktop which predicts the next word that is useful for document writing and email writing. The predicted word may not be appropriate it depends on the corpus trained. The existing system doesn't check the words before it adds in to the dictionary. Though there are many text editors but there is no module developed for sentence correction or prediction based on grammatical rules. Though they are many mobile applications which suggests next word and auto corrects spellings, they are not embedded in single application along with correction of sentence.

Methodology

As shown in Figure1, User inputs the text word by word. Input is sent through text extraction unit where the text is tokenised and pre-processed. Pre-processed data is sent through three modules, Auto correct engine where the spelling of the word is verified with the dictionary. Auto complete module where the input words are sent through bigram model to predict the next possible word using corpus trained. Sentence Correction module checks for the syntactic structure of the sentence using grammatical rules.

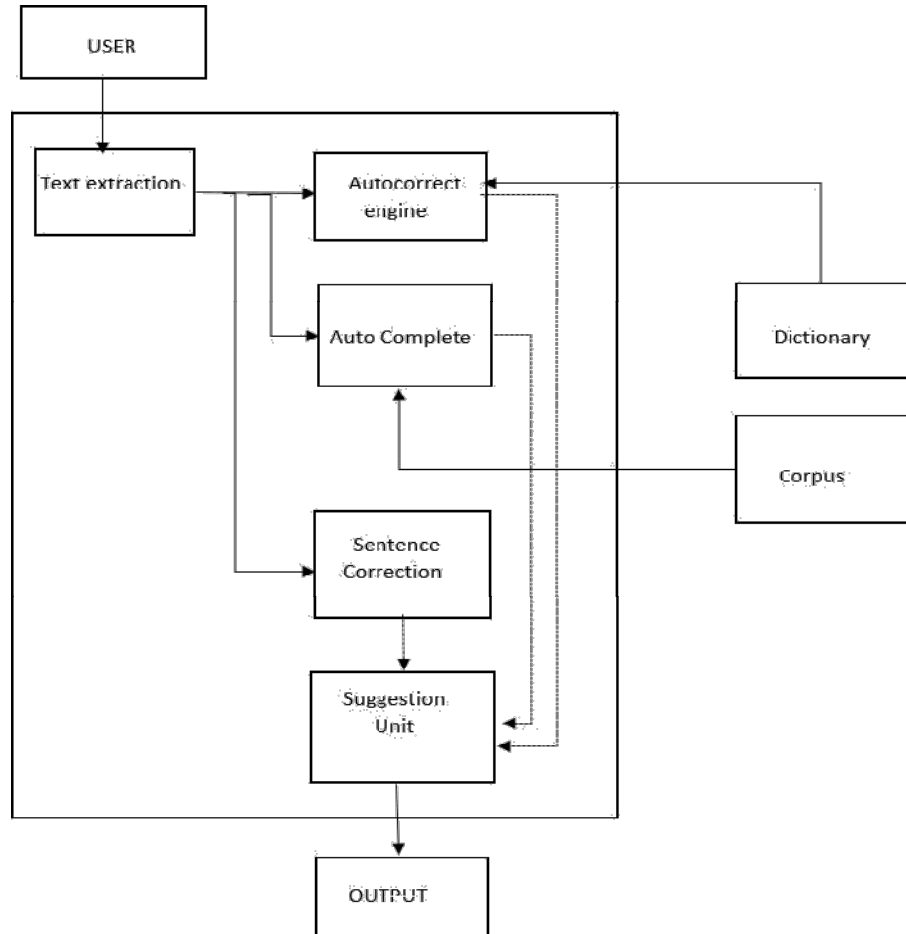


Figure 1: Flowchart Depicting Design Steps for Proposed Text Prediction

Text Prediction

Text Prediction is achieved by following a series of steps. The steps followed are mentioned below:

1. Pre-processing of raw data
2. Spell Check Model
3. N-gram model using corpus text
4. Sentence Correction

Steps in Pre-processing

1. Collect the required data
2. Read the data
3. Convert the data into UTF-8 format
4. Split the data into tokens
5. Remove unnecessary characters other than text
6. Divide the token into trigrams
7. Assign value key pairs to tokens
8. Store them in hash map data structure

Procedure for spell check

Install the package or the library **enchant** [10]

Input the word that is to be checked

If word given as input exactly matches with the word in dictionary it is returned

Else the word that nearly matches with the words in dictionary is appended to the empty dictionary initialized.

Word is compared to each word in the dictionary created

If any insertions, deletions or manipulations needed, those operations are performed

Returns the correct spelling word

Procedure for N gram model

Last two words from input are identified

These words are sent to trigram model (naming the parameters as current and next)

These two words are together given a key.

This key compared with the keys in dictionary that are stored during preprocessing.

If key exists that value is appended to a empty dictionary

Return the word which has the highest value i.e. the most repeated trigram and expected next word based on the corpus trained.

Procedure for Sentence Correction

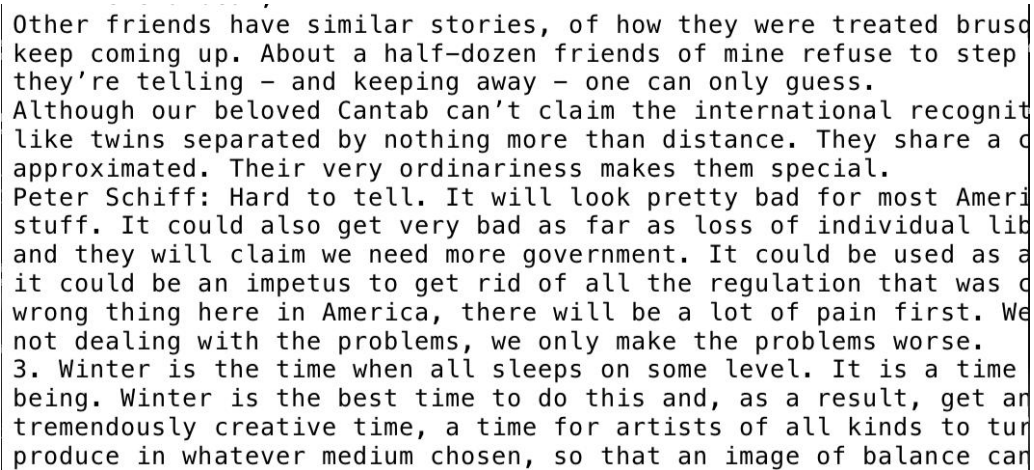
1. Enter a sentence or paragraph.
2. Perform spelling check and correction.
3. Categorize sentence in eight different types.
4. Perform part of speech tagging.
5. Parse the sentence using grammar rule.
6. Form parse tree [5] If parse tree is not generated, then report that the sentence is incorrect else correct.

A sentence is given as input to sentence correction module. The sentence is tokenized and the words are appended to the list, applying Parts of Speech tagger to the words in the list. Store these tags in lists respectively to the input sentence. Now these tags are sent to the grammar to check whether the given sentence structure is syntactically correct or not according to the grammatical rules. This module uses Chart parser which compares the rules determined if matches with any produces a parse which represents given sentence is syntactically correct according to grammatical rules. This sentence correction is done in dynamic approach, data is divided into chunks. It doesn't uses backtracking mechanism which leads to infinite loop which is hard to resolve.

Results & Discussion

Data Set Preparation/Collection

The data set for Text prediction is large corpus taken from US blogs and news with well defined English. Any data set can be considered which has well defined English. The data set we have considered is around 210mb.



Other friends have similar stories, of how they were treated brusque keep coming up. About a half-dozen friends of mine refuse to step they're telling - and keeping away - one can only guess. Although our beloved Cantab can't claim the international recognition like twins separated by nothing more than distance. They share a close approximated. Their very ordinariness makes them special. Peter Schiff: Hard to tell. It will look pretty bad for most American stuff. It could also get very bad as far as loss of individual liberty and they will claim we need more government. It could be used as a it could be an impetus to get rid of all the regulation that was a wrong thing here in America, there will be a lot of pain first. We not dealing with the problems, we only make the problems worse. 3. Winter is the time when all sleeps on some level. It is a time being. Winter is the best time to do this and, as a result, get an tremendously creative time, a time for artists of all kinds to turn produce in whatever medium chosen, so that an image of balance can

Figure 2: Sample Corpus Text

The corpus is pre-processed and tokenized. The tokenized words are stored and retrieved with the dictionary inside a dictionary data structure, hash map implementation. Time and space complexities are reduced with hash map explanation.

```
(projenv) Adithis-MacBook-Pro:sampletesting adithiloka$ python testpro.py
enter of his
1.19209289551e-05
money
(projenv) Adithis-MacBook-Pro:sampletesting adithiloka$ python ngram.py
enter : of his
[(u'money', 1)]
[(u'money', 1)]
0.00112795829773
money
```

Figure 3: Variation in Time Complexities

In figure3, it depicts the variation in time complexities with and without hash map implementation. Testpro.py [11] does the linear search which takes 1.19209289551e-05 amount of time. Ngram.py [11] has the hash map implementation which takes 0.00112795829773 amount of time which is relatively very less when compared to linear search. This increases the efficiency of storage and retrieval of the words.

1. Spell Check

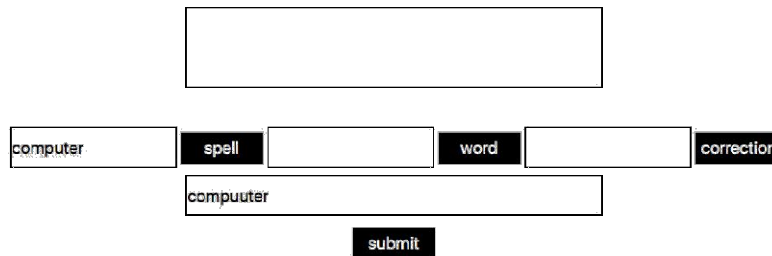


Figure 4: Screenshot of Spell check in Text Prediction

Figure 4 spell check depicts spelling correction module in Text prediction. Given input word is **computer** which is actually a wrongly spelt word. With few deletions in the given input it returned the correct spelling as **computer**.

2. N gram Model

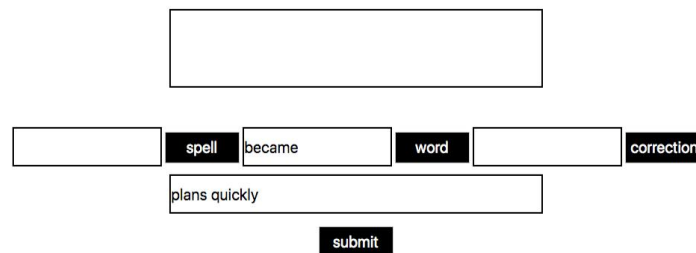


Figure 5: Screen Shot of N gram Model of Text Prediction

In Figure 5, depicts the next word prediction using n gram model based on the corpus trained. 'plans quickly became' is the most repeated trigram so it has returned became.

3. Sentence Correction

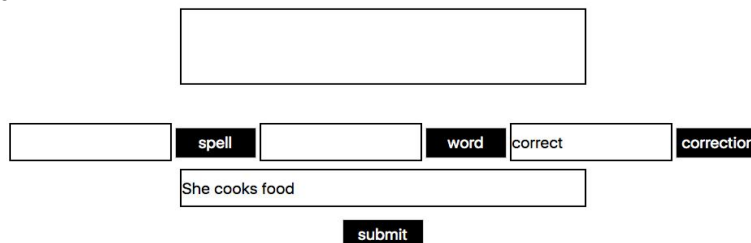


Figure 6: Screenshot of Sentence Correction Module of Text Prediction

In Figure 6, sentence correction module of text prediction checks for the syntactic structure of the sentence taken as input according to the grammatical rules and returns whether the given input sentence structure is correct or not. Here, 'she cooks food' being the input sentence is syntactically correct so the output is correct.

4. Final Output

The complete text prediction is explained in the following Figure7.

Figure 7: Complete Output of Text Prediction

Analysis of Result

1. Spell Check

Almost 6 out of 8 results have given the correct output. Few outputs may give the first most matched word rather than the expected word. For example in Table 1 gamng is expected to be gaming but gang is first encountered by deletion of an extra m letter.

Table 1: Results of Spell Check Module

Computerrr	Computerize
attacment	attachment
begininggg	Beginning
atmosphere	atmosphere
specialiy	Speciality
multplicity	multiplicity
economical	economical
gamng	Gang

2. N-gram Model Plot

The pictorial representation of frequency of words in a sample corpus text taken using matplotlib is shown in following Figure 8 and Figure 9.

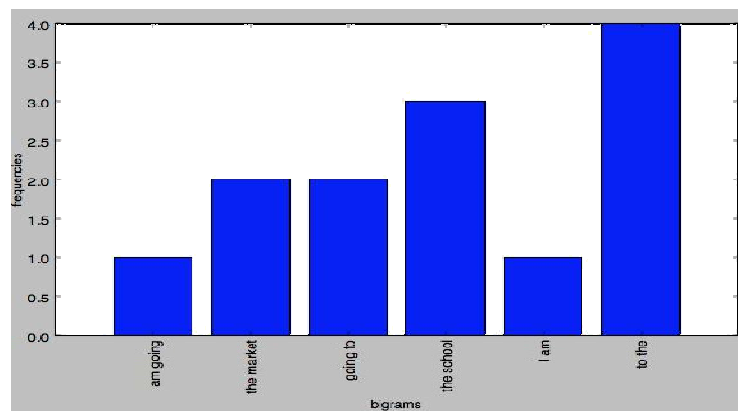


Figure 8: Bar Graph Plot for Frequency of Bigrams in a Corpus

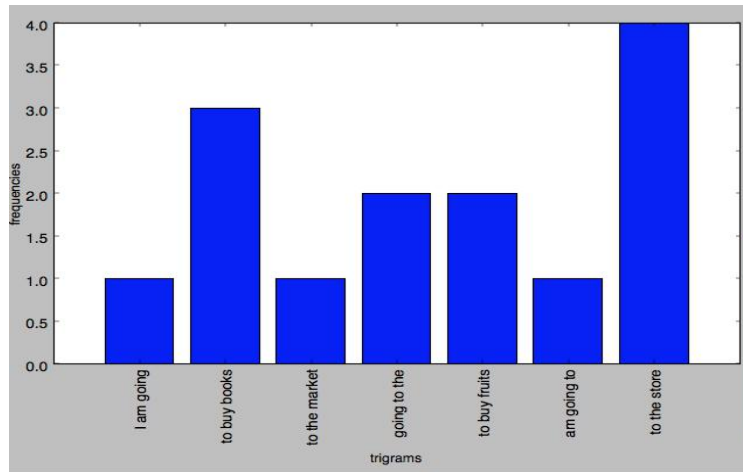


Figure 9: Bar Graph Plot for Frequency of Trigrams in a Corpus

3. Sentence Correction

To an extent, rules are framed in such a way to overcome few ambiguities. Table 2 determines the structure of few sentences.

Table 2: Results of Sentence Correction Module for Various Sentences

She likes him	Correct
She like him	Incorrect
Ram is playing	Correct
I have doing this	Incorrect
She is short then him	Incorrect
She is short than him	Correct
She ate an apple	Correct
I have been doing this	Correct

4. Performance Measure

Accuracy is not really a reliable metric for the real performance of a classifier when the number of samples in different classes vary greatly (unbalanced target) because it will yield misleading results. For example, if there were 95 cats and only 5 dogs in the data set, the classifier could easily be biased into classifying all the samples as cats. The overall accuracy would be 95%, but in practice the classifier would have a 100% recognition rate for the cat class but a 0% recognition rate for the dog class. Overall accuracy is calculated as:

As in the proposed text prediction system based on the rules given the given sentence is classified as either correct or incorrect. For this there is no such data set that is used for training. It has grammatical rules programmed which check the correctness of structure of sentence. The algorithm gave an accuracy of 81% for the test data. Following is the hypothesis we considered for the model:

Table 3: Confusion Matrix Values for 100 Observations

P/N	Positive	Negative
Positive	41	9
Negative	10	40

Conclusion

The developed model is a web application which corrects the spellings of the text entered and predicts the next word based on two previous words using n gram model. And the model checks for the syntactic structure of the sentence using grammatical rules. Till now, there is no such application with spelling correction, word prediction and sentence correction in a single application. In addition, tried to improve the efficiency of the system by implementing in hash map data structure. Time and space complexities have reduced.

Future Work

To an extension of this project, sentence correction can be done. Suggesting a appropriate sentence according to the context and grammatical rules is complex. Instead of single corpus, when multiple corpus is given pre-processing takes much time. Parallel processing can be done to reduce the search and pre-processing of the corpus initially.

References

- [1].Akshay Bhatia , Amit Bharadia, Kunal Sawant, Swapnali Kurhade , “Predictive and Corrective Text Input for desktop editor using n-grams and suffix trees” at International Conference on Advances in Human Machine Interaction (HMI - 2016),March 03-05, 2016, R. L. Jalappa Institute of Technology, Doddaballapur, Bangalore, India.
- [2].S.Satapathy, K.Asnani, D.Vaz, T.PrabhuDesai, S.Borgikar, M.Bisht, S.Bhosale and N.Balaji, Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: “Theory and Applications” (FICTA). Cham [u.a.]: Springer, 2015, pp. 397-404.
- [3].Madhuri A.Tayal, Dr. M. M. Raghuwanshi ,Dr. Latesh Malik “Syntax Parsing: Implementation using Grammar-Rules for English Language” at International Conference on Electronic Systems, Signal Processing and Computing Technologies 2014.
- [4].Bharti Akshar and Rajeev Sangal, “A Karaka-based approach to parsing of Indian languages”, Proceedings of the 13th Conference on Computational Linguistics, Association for Computational Linguistics.
- [5].Chung-Hsien Wu, Senior Member, IEEE, Chao-Hong Liu, Matthew Harris, and Liang-Chih Yu, “Sentence Correction Incorporating Relative Position and Parse Template Language Models” at ” at International Conference on Advances in NLP,August 2010.
- [6].C.Wren and H.Martin. “High School English Grammar and Composition”.
- [7].Infante-Lopez, Gabriel and Maarten de Rijke, “A note on the expressive power of probabilistic context free grammars”, Journal of Logic, Language and Information, Kluwer Academic publisher, 15 (3), 2006.
- [8].Bottle framework documentation, Source available at <https://bottlepy.org/docs/dev/tutorial.html>
- [9].Spell checking library called PyEnchant, Source available at <http://pyenchant.readthedocs.io/en/latest/api/enchant.checker.html>
- [10].Natural language Processing pre-processing steps, Source available at <http://www.nltk.org/book/ch01.html>
- [11].Visualization with Matplotlib, Source available at <https://anaconda.org/ijstokes/16-visualization-matplotlib/notebook>
- [12].“Head-Driven Statistical Models for Natural Language Processing” by Michael Collins,1999.
- [13].Namrata Pratap Simha, Vishwas Manohar, Sudarshan Suresh M., Dheeraj D. Bhat, Dr. Saritha Chakrasali “Rule based Simple English Sentence Correction by Rearrangement of Words” at International Journal of Scientific & Engineering Research, Volume 7, Issue 6, June-2016.
- [14].Yanen Li, Huizhong Duan and Cheng Xiang Zhai “A Generalized Hidden Markov Model with Discriminative Training for Query Spelling Correction” at University of Illinois at Urbana-Champaign, Urbana.