International Journal of Computer Applications Technology and Research
Volume 3– Issue 12, 812 - 814, 2014, ISSN:- 2319–8656

Sentence Validation by Statistical Language Modeling and Semantic Relations

Lakshay Arya
Guru Gobind Singh Indraprastha University
Maharaja Surajmal Institute Of Technology
New Delhi, India

Abstract : This paper deals with Sentence Validation - a sub-field of Natural Language Processing. It finds various applications in different areas as it deals with understanding the natural language (English in most cases) and manipulating it. So the effort is on understanding and extracting important information delivered to the computer and make possible efficient human computer interaction. Sentence Validation is approached in two ways - by Statistical approach and Semantic approach. In both approaches database is trained with the help of sample sentences of Brown corpus of NLTK. The statistical approach uses trigram technique based on N-gram Markov Model and modified Kneser-Ney Smoothing to handle zero probabilities. As another testing on statistical basis, tagging and chunking of the sentences having named entities is carried out using pre-defined grammar rules and semantic tree parsing, and chunked off sentences are fed into another database, upon which testing is carried out. Finally, semantic analysis is carried out by extracting entity relation pairs which are then tested. After the results of all three approaches is compiled, graphs are plotted and variations are studied. Hence, a comparison of three different models is calculated and formulated. Graphs pertaining to the probabilities of the three approaches are plotted, which clearly demarcate them and throw light on the findings of the project.

Keywords: language modeling, smoothing, chunking, statistical, semantic

1. INTRODUCTION

NLP is a field of Computer Science and linguistics concerned with interactions between computers and human languages. NLP is referred to as AI-complete problem. Research into modern statistical NLP algorithms require understanding of various disparate fields like linguistics, computer science, statistics, linear algebra and optimization theory.

To understand NLP, we have to keep in mind that we have several types of languages today : Natural Languages such as English or Hindi, Descriptive Languages such as DNA, Chemical formulas etc, and artificial languages such as Java, Python etc. We define Natural Language as a set of all possible texts, wherein each text is composed of sequence of words from respective vocabulary. In essence, a vocabulary consists of a set of possible words allowed in that language. NLP works on several layers of language: Phonology, Morphology, Lexical, Syntactic, Semantic, Discourse, Pragmatic etc. Sentence Validation finds its applications in almost all fields of NLP - Information Retrieval, Information Extraction, Question-Answering, Visualization, Data Mining, Text Summarization, Text Categorization, Machine and Language Translation, Dialogue And Speech based Systems and many other one can think of.

Statistical analysis of data is the most popular method for applications aiming at validating sentences. N-gram techniques make use of Markov Model. For convenience, we restrict our study till trigrams which are preceded by bigrams. Results of this approach are compared with results of Chunked-Off Markov Model. Extending our study and moving towards Semantic Analysis - we find out the Entity-Relation pairs from the Chunked off bigrams and trigrams. Finally, we aim to calculate the results for comparison of the three above models.

2. SENTENCE VALIDATION

Sentence validation is the process in which computer tries to calculate the validity of sentence and gives the cumulative probability. Validation refers to correctness of sentence, in dimensions such as statistical and semantic. A good validation program can verify whether sentence is correct at all levels.

Python language and its NLTK [5] suite of libraries is most suited for NLP problems. They are used as a tool for most of NLP related research areas - empirical linguistics, cognitive science, artificial intelligence, information retrieval and machine learning. NLTK provides easily-guessable method names for word tokenizing, sentence tokenizing, POS tagging, chunking, bigram and trigram generation, frequency distribution, and many more. Oracle connectivity with Python is used to store the bigrams, trigrams and entity-relation pairs required to test the three different models and finally to compare their results.

First model is the purely statistical Markov Model, i.e. bigrams and trigrams are generated from the sample files of Brown corpus of NLTK and then fed into the database. Testing yields some results and raises some disadvantages which will be discussed later. Second model is Chunked-Off Markov Model - an extension of the first model in the way that it makes use of tagging and chunking operations wherein all the proper nouns are categorized as PERSON, PLACE, ORGANIZATION, FACILITY, etc. This replacing solves some issues which purely statistical model could not deal with. Moving from statistical to semantic approach, we now aim to validate a sentence on semantic basis too, i.e. whether

the sentence has some meaning and makes sense or not. For example, 'PERSON eats' is a valid sentence whereas 'PLACE eats' is an invalid one. So the latter input sentence must result in a low probability for correctness compared to the former. In order to show this demarcation between sentences, we extract the entity relation pairs from sample sentences using named entity recognition and chunking and store them in the ER database. Whenever a sentence comes up for testing, we

extract the E-R pairs in this sentence and match them from database entries to calculate probability for semantic validity.

The same corpus data and test data for the above three approaches are taken for comparison purposes. Graphs pertaining to the results are plotted and major differences and improvements are seen which are later illustrated and analyzed.

3. HOW DOES IT WORK ?

The first two statistical approaches use the N-gram technique and Markov Model[2] building. In the pure statistical Markov N-gram Model, corpus data is fed into the database in the form of bigrams and trigrams with their respective frequencies(i.e. how many times they occur in the whole data set of sample sentences). When an input sentence is to be validated, it is tokenized into bigrams and trigrams which are then matched with database values and a cumulative probability after application of Smoothing-off technique of Kneser-Ney Smoothing which handles new words and zero count events having zero probability which may cause system crash, is calculated.

Chunked-Off Markov Model makes use of our own defined replace function implemented through pos_tag and ne_chunk functionality of NLTK. Every sentence is first tagged according to Part-Of-Speech using pos_tag. Whenever a 'NN', 'NNP' or in general 'NN*' chunk is encountered, it is passed to ne_chunk which replaces the named entity with its type and returns a modified sentence whose bigrams and trigrams are generated and fed into the database. The testing procedure of this approach follows above methodology and modifies the sentence entered by the user in the same way, calculates the probabilities of the bigrams and trigrams by matching them with database entries and finally smoothes off to yield final results.

Above two approaches are statistical in nature, but we need to validate sentences on semantic and syntactic basis as well, i.e. whether sentences actually make sense or not. For bringing this into picture, we extract all entities(again NN* chunks) and relations(VB* chunks). We define our own set of grammar rules as context free grammar to generate parse tree from which E-R pairs are extracted.

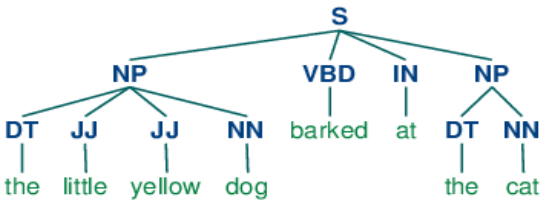

Figure. 1 Parse Tree Generated by CFG

4. COMPLETE STRUCTURE

We have trained the database with 85% corpus and testing with the rest of 15% corpus we have. This has two advantages - firstly we shall use the same ratio in all other approaches so that we can compare them easily. Secondly it provides a threshold value for probability which will help us to distinguish between correct and incorrect test sentences depicting regions above and below threshold respectively. Graphs are plotted between probability(exponential, in order of 10) and length of the sentence(number of words).

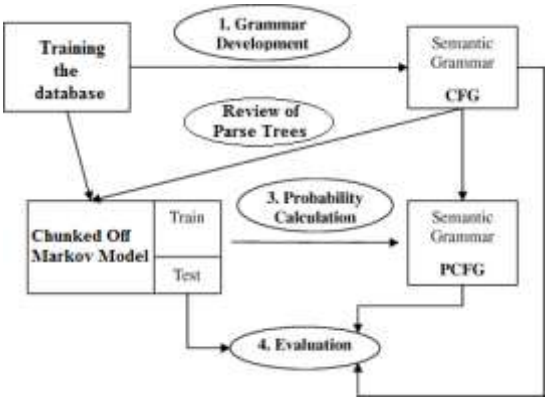

Figure. 2 Complete flowchart of Sentence Validation process

4.1 N-Gram Markov Model

The first module is Pure Markov Model[1]. In the pure statistical Markov N-gram Model, corpus data is fed into the database in the form of bigrams and trigrams with their respective frequencies(i.e. how many times they occur in the whole data set of sample sentences). When an input sentence is to be validated, it is tokenized into bigrams and trigrams which are then matched with database values and a cumulative probability after application of Smoothing-off technique of Kneser-Ney Smoothing is calculated. The main disadvantage of this pure statistics-based model is that it is not able to deal with Proper Nouns and Named Entities. Whenever a new proper noun is encountered with the same relation, it will result in lower probability even though the sentence might be valid. This shortcoming of Markov Model is overcome by next module - Chunked Off Markov Model. Markov Modeling is the most common method to perform statistical analysis on any type of data but it cannot be the sole model for testing of NLP applications.

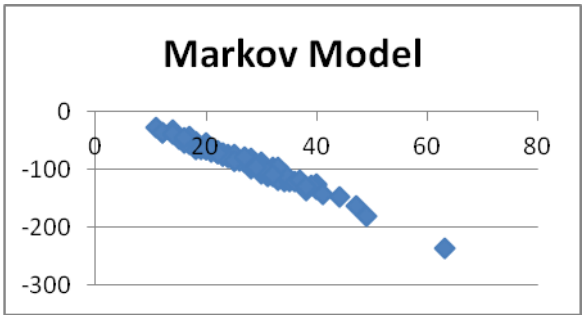

Figure. 3 Testing results for Pure Statistical Markov Model

4.2 Chunked-Off Markov Model

The second module is Chunked-Off Markov Model[3] - training the database with corpus sentences in which all the nouns and named entities are replaced with their respective type. This is implemented using the tagging and chunking operations of NLTK. This solves the problem of Pure Statistical model that it is not able to deal with proper nouns. For example, a corpus sentence has the trigram 'John eats pie'. If a test sentence occurs like 'Mary eats pie', it will result in a very low trigram probability. But if the trigram 'John eats pie' is modified to 'PERSON eats pie', it will result in a better comparison.

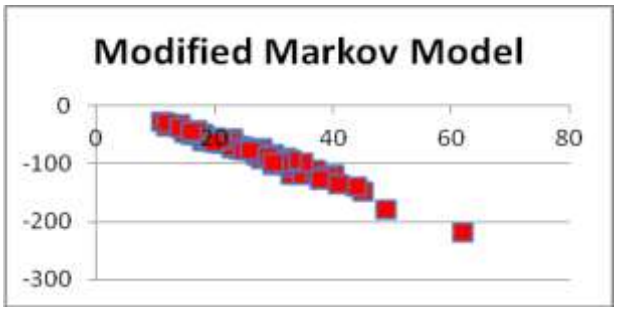

Figure. 4 Testing results for Chunked-Off Markov Model

4.3 Entity-Relation Model

The third module is E-R model[4]. Extraction of the entities(all NN* chunks) and relations(all VB* chunks). We define a set of grammar rules as context free grammar to generate parse tree from which E-R pairs are extracted and entered into the database. For convenience, we have taken the first main entity and the first main relation because compound entities are difficult to deal with.

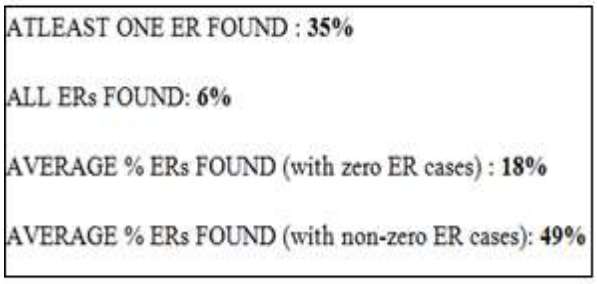

Figure. 5 Testing results for E-R Model

4.4 Comparison of the three models

The fourth module is comparison. As expected, the modified Markov Chunked-Off model performs. We can also see that there are no sharp dips in the modified model which are present in pure statistical model due to a sharp decrease in the probability of trigrams and bigrams. The modified model is consistent due to its ability to deal with Proper Nouns and Named Entities.

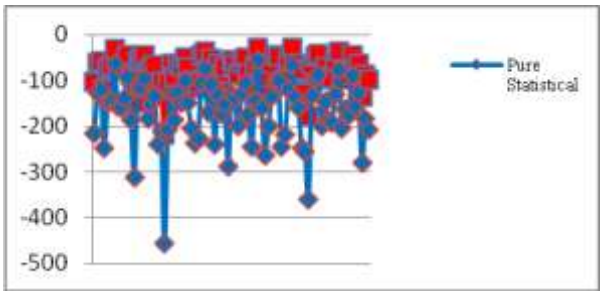

Figure. 6 Comparison of the two models

5. WHY SENTENCE VALIDATION ?

Sentence Validation finds its use in the following fields :-

1. Information systems
2. Question-answering systems
3. Query-based information extraction systems
4. Text summarization applications
5. Language and machine translation applications
6. Speech and Dialogue based systems

For example, Wolfram-Alpha, Google Search Engine, Text Compactor, SDL Trados, Siri, S-Voice, etc all integrate sentence validation as an important module.

6. CHALLENGES AND FUTURE OF SENTENCE VALIDATION

As mentioned earlier NLP is still in the earliest stage of adoption. Research work in this field is still emerging. It is a difficult task to train a computer and make it understand the complex and ambiguous nature of natural languages. The statistical approach is a well proven approach for statistical calculations. But the data obtained from ER approach is inconclusive. We may have to improve our approach and scale the data to make ER model work. ER Model offers very substantial advantages over Statistical Model, that makes this approach worth looking into. Even if it cannot reach the levels of Markov Model, ER Model could be a powerful tool in complementing Markov Model as well as for variety of other NLP Applications.

We see Sentence Validation as the single best method available to process any Natural Language application. All languages have own set of rules which are not only difficult to feed in a computer, but are also ambiguous in nature and complex to comprehend and generalize. Thus, different approaches have to be studied, analyzed and integrated for accurate results.

Our three approaches validate a sentence in an over-all manner, both statistically and semantically, making this system an efficient one. Also, the graphs show clearly that chunking of the training data will yield in better testing of data. The testing will become even more accurate if database is expanded with more sentences.

7. REFERENCES

[1] Chen, Stanley F. and Joshua Goodman. 1998 An empirical study of smoothing techniques for language modeling Computer Speech & Language 13.4 : 359-393.

[2] Goodman. 2001 A bit of progress in Language Modeling

[3] Rosenfield, Roni. 2000 Two decades of statistical language modeling : Where do we go from here?

[4] Nguyen Bach and Sameer Badaskar. 2005 A Review of Relation Extraction

[5] NLTK Documentation. 2014 Retrieved from http://www.nltk.org/book