Natural Language Processing for Sentence Completion Tasks

Adithya Addanki
Department of Computer Science,
The University of Akron,
Akron, Ohio-44304, aa207@zips.uakron.edu

Srinivasa Rao Katta
Department of Computer Science,
The University of Akron,
Akron, Ohio-44304, sk189@zips.uakron.edu

Abstract

In the field of Natural language processing, sentence completion using semantic coherence is the biggest challenge. This paper discusses different n-gram models (both forward and backward) and Latent Semantic Analysis model, including a comparison study that explains where these models face challenges. MSR Sentence Completion Challenge Data, which consists of 1,040 sentences has been used to validate the models on their accuracy in predicting the right word for the sentence completion tasks. Finally, we discuss various insights gained on the models and observations based on their prediction accuracy.

1. Introduction

The primary goal of this project is to automatically answer sentence completion questions using Natural Language Processing techniques. To understand semantic coherence on a sentence level we study different N-Gram models and Latent Semantic Analysis Model. We validated the models using fill-in-the-blank questions similar to those found on the widely used Scholastic Aptitude Test. The sentence completion questions focus on testing the student's ability to select words which are meaningful and coherent with the sentence. This determination cannot be made on the basis of grammatical correctness alone, it requires understanding of sentence structures, syntax and semantics, and linguistics of English language.

The training set consists of 522 19th century novels from Project Gutenberg, the same training set suggested by Microsoft Research Sentence Completion challenge. The testing set consists of 1,040 sentences from five Sherlock Holmes novels by Sir Arthur Conan Doyle [4]. In each of these sentences, an infrequent word is chosen as the focus of the question, which has to be deduced by our models.

2. Background

2.1 N-Gram Models

N represents the number of words to be considered for the sentence completion task, when N=2 (bigram), the word preceding or succeeding the blank is taken into consideration for backward and forward models respectively. A probability of an N-gram is calculated based on the estimation that the words preceding or succeeding the blank have been observed in the corpus in the same order.

2.2 Smoothing

Smoothing is a mechanism that helps deal with previously unseen N-grams, so that there is no bias during calculation of probability of occurrence.

Laplace Smoothing: We consider the frequency of previously unseen N-gram as 1 against the total number of words in the dataset. Thus not neglecting the first occurrence of the N-Gram.

Backoff Model: We start with the highest N-Gram model and fall back to the lower gram (N-1) model in case the probability of the highest is zero.

2.3 Forward and Backward Probability

Backward probability is defined as the conditional probability that the target option word W_i will occur given the earlier word sequences W_{i-1}, W_{i-2},….W_1. Similarly, forward probability is defined as the conditional probability that target option word W_i will occur given the later word sequences W_{i+1},W_{i+2}, ...W_n.

2.4 Latent Semantic Analysis

Latent Semantic Analysis is a widely accepted and well known approach to represent words and documents in a reduced vector space. A matrix representing the counts of each of the words in the vocabulary, in each document is constructed and Singular Value Decomposition is applied. The resulting orthonormal vectors of words and documents are used to extract the features of similarities in between words that occur in an analogous context. The similarity between two words could be calculated as the cosine similarity between the scaled vectors [6]. The rows of the reduced orthonormal vectors have the same behavior as the original vectors in the original space. The similarity between the blank word and the rest of the words in the sentence is considered as a measure for deducing the right word [5].

3. Dataset Collection

Training Dataset: Initially, we obtained our training dataset from Microsoft Projects website [1]. This dataset consists of 522 nineteenth century novels. After building the models, midway through validation, we found this data to be inadequate and was yielding low prediction accuracies as the models were trained based on the number of unique words in the complete data set. In an attempt to get a superior corpus to train our model, we requested access to the Microsoft web n-gram corpus [2]. This is a web scale dataset, with everyday updates, and can be accessed through REST or SOAP web service [3].

Testing Dataset: MSR sentence completion challenge testing set, also considered as the gold standard is used. The answers to these 1040 sentences are already available, the testing set along with the answers could be downloaded from Microsoft server [1].

3.1 Data Preprocessing

All the 1040 testing sentences are converted to the below format and are stored in a questions file and corresponding answers are stored in an answer file. All the models used in this project are run on this preprocessed data. While traversing through the question and answer files, the data is further normalized by removing unwanted characters like delimiters, extra spaces and numerals for making the grammar regular.

Question: I have seen it on him, and could ______ to it.

A. write

B. migrate

C. climb

D. swear

E. contribute

The above question is converted to a format with imposter words in the place of blank.

1) I have seen it on him, and could [write] to it.
2) I have seen it on him, and could [migrate] to it.
3) I have seen it on him, and could [climb] to it.
4) I have seen it on him, and could [swear] to it.
5) I have seen it on him, and could [contribute] to it.

4. Language Models

4.1 Standard Bigram Language Model

The standard bigram language model constructs the bigrams set by traversing through the training data. While testing, model reads five sentences from the questions file at a time and construct bigrams from each sentence. Then, it calculates the bigram probability (the imposter word and the word preceding or succeeding it W_i & W_{i-1}) of the sentence. The option in the sentence with the highest probability is considered as the answer. To deal with previously unseen bigrams, we used Laplace smoothing. This model produced a prediction accuracy of **20.77 %**. The prediction accuracy increased to **26.05 %** when we considered only the bigrams with a frequency greater than 20.

4.2 Standard Higher Order N-gram Language Model

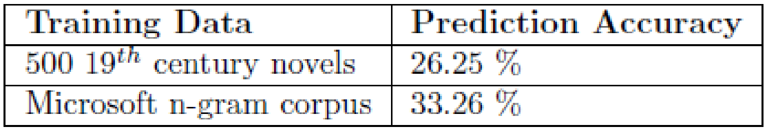
The probability is calculated for trigram - W_i; W_{i-1}; W_{i-2} where W_i is the option word. This model resulted in a prediction accuracy of 44.33 %. Similar process is applied to fourgram model - W_i; W_{i-1}; W_{i-2}; W_{i-3} where W_i is the option word. This model resulted in a prediction accuracy of 64.13 %

Bigrams Type	**Prediction Accuracy**
Trigrams	44.33%
Fourgrams	64.13%

Table 4.2.1: Trigram and Fourgram Models Accuracy

4.3 Forward-Backward Bigram Language Model

The forward and backward probabilities are summed up for each of the 5 answer sentences and the sentence having the maximum probability among the five sentences for a given question is selected as the correct answer. This was done based on the earlier work of Kyusong Lee & Gary Geunbae Lee, 2014 which is the present state of the art in sentence completion task [5]. When this model is trained on the 522 19th century novels, it gave a prediction accuracy of **26.25%.** On changing the training data to the Microsoft n-gram corpus, the prediction accuracy increased to **30.12 %.**



Training Data	Prediction Accuracy
500 19th century novels	26.25 %
Microsoft n-gram corpus	33.26 %

Table 4.3.1: Forward Backward Bigram Model Accuracy

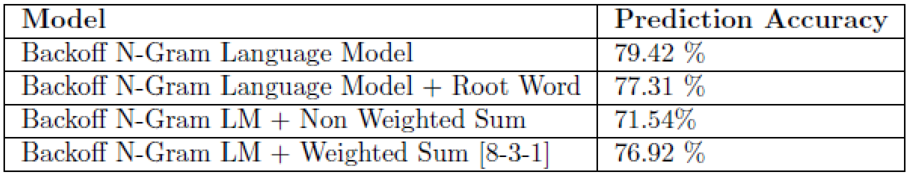
4.4 Backoff N-gram Language Model

Backoff model is used which falls back to lower order n-grams when the probabilities of occurrence of the higher order n-grams are zero. Using this model, a prediction accuracy of **79.42 % is** obtained**.**

Root Word: First of all, we incorporated the concept of root words into the model. For this, we used WordNet tool and the Java WordNet Interface API [7] to retrieve the root word for every actual option word in all of the sentences. Then, the probabilities using the actual option word in the n-gram and the probabilities using the root word in the n-gram are summed up for each of the 5 answer sentences and the sentence having the maximum probability among the five sentences for a given question is selected to be having the correct answer. In case the root word is same as the actual option word, the latter probability is taken as zero. This hybrid model, has a reduced prediction accuracy by roughly 2 % when compared to the above model. The prediction accuracy was observed to be **77.31 %.**

Weighted N-grams: Used a weighted sum as feature for predicting the correct answer. When we gave weights of 4, 3 and 2 to fourgrams, trigrams and bigrams respectively, the prediction accuracy was **74.6 %.** When we altered the weights to 8, 3 and 1 for fourgrams, trigrams and bigrams respectively, the prediction accuracy increased to **76.92 %.**

Similarly, when non-weighted sum of all order n-grams are used, the prediction accuracy was observed as **71.54 %**



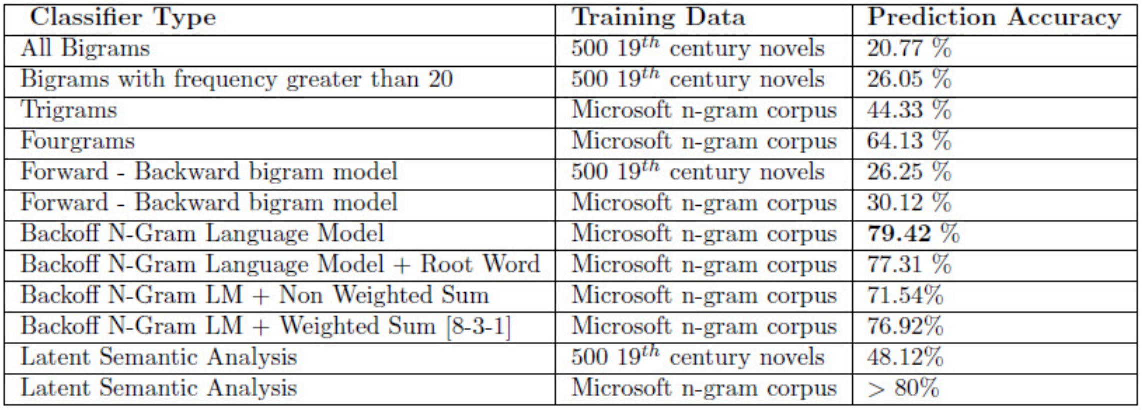
Model	Prediction Accuracy
Backoff N-Gram Language Model	79.42 %
Backoff N-Gram Language Model + Root Word	77.31 %
Backoff N-Gram LM + Non Weighted Sum	71.54%
Backoff N-Gram LM + Weighted Sum [8-3-1]	76.92 %

Table 4.4.1: Backoff N-gram Model Accuracy

4.5 Latent Semantic Analysis (LSA)

As discussed before Latent Semantic Analysis incorporates finding the inherent salient features and relationship between words and documents classified according to the genre. A matrix is constructed with word frequencies for each document; since the matrix obtained is sparse, we constructed the matrix with unique words occurring only in the testing dataset and Singular valued decomposition (SVD) is performed on this matrix [6]. After obtaining word vector, document vector with the singular values in diagonal matrix, word similarity for the target word with the rest of the words in the sentence is calculated for each of the five candidate sentences. The one which is having highest value will be considered as correct answer to the question. Unfortunately, we couldn't implement this on the Microsoft web corpus, as the rest web service returns only the probability of a word rather than the word frequency per document. Prediction accuracy of **48.12%** is obtained by the use of trigrams on the novels dataset which is a good **4%** increase.

To summarize all our results thus far,



Classifier Type	Training Data	Prediction Accuracy
All Bigrams	500 19^{th} century novels	20.77 %
Bigrams with frequency greater than 20	500 19^{th} century novels	26.05 %
Trigrams	Microsoft n-gram corpus	44.33 %
Fourgrams	Microsoft n-gram corpus	64.13 %
Forward - Backward bigram model	500 19^{th} century novels	26.25 %
Forward - Backward bigram model	Microsoft n-gram corpus	30.12 %
Backoff N-Gram Language Model	Microsoft n-gram corpus	**79.42** %
Backoff N-Gram Language Model + Root Word	Microsoft n-gram corpus	77.31 %
Backoff N-Gram LM + Non Weighted Sum	Microsoft n-gram corpus	71.54%
Backoff N-Gram LM + Weighted Sum [8-3-1]	Microsoft n-gram corpus	76.92%
Latent Semantic Analysis	500 19^{th} century novels	48.12%
Latent Semantic Analysis	Microsoft n-gram corpus	> 80%

Table 4.1: Model Accuracies

Implementing latent semantic analysis on a corpus as large as the Microsoft n-gram web corpus, the accuracy is likely to improve around **3-4% (>80%).**

5. Error Analysis

Error analysis is given below to observe the specific cases that are failing and the cause of failure to predict the right option.

i. The first observation is that identifying the word sequence occurring with high probability is not sufficient. Capturing the relationship between the target blank and the surrounding words in the sentence may help us to infer about the correct option to be filled in the blank in a better way.

Question: They seize him and use violence towards him in order to make him sign some papers to make over the girl's ________ of which he may be trustee to them.

A. appreciation

B. activity

C. suspicions

D. administration

E. fortune

For the above question the model assumed option D- administration to be correct while in fact E -fortune is the correct option. Although the model takes into account the frequency of the bigram "**girl’s administration**" for instance, it fails to take into account the presence of semantically related words in the sentence like "**trustee**" to give more weightage to the correct bigram "**girl’s fortune**".

ii. Also an observation made that in cases where blank is positioned at first or last index of the sentence, the answer is predicted to be wrong consistently. An example of this kind of question would be

Question: _________ by nature, Jones spoke very little even to his own family members.

A. garrulous

B. equivocal

C. taciturn

D. arrogant

E. gregarious

This is because either backward or forward probability features are missing and remaining few probability features are unable to give weightage to the answer sentences with the correct option.

6. Comparison Study

A brief comparison of the models helped us observe the following differences.

Window of Immediate Context vs Local and Logical Coherence: N-Gram models give a prediction based only on certain predetermined length of sequence of words in a particular order, whereas the LSA model deduces the word based on the context and words present in the sentence; LSA is truly a bag of words model [6].

Coverage: N-Grams is lower as it takes only a window of words into consideration compared to LSA which takes the whole sentence and its structure into consideration thus extracting a semantic feature.

Syntactic and Semantic Dependencies: N-Gram model fails to identify the syntactic and semantics of English language on the whole, thereby leading to unrealistic answering of questions. N-Gram is based on word extraction where as LSA is feature extraction based.

Artificial Horizon: N-grams require a parameter; N to be specified for which the frequency has to be retrieved for calculating the probability, thus making it an artificial horizon.

Relationship between Words and Documents: LSA model establishes a salient feature oriented relationship between words and documents in the entire dataset, clustering them in a reduced vector space [6]. N-Gram model establishes no relationship between words and documents in the dataset.

7. Observations and Insights

7.1 Observations

During the course of the project we have come across many interesting observations.

i. In the standard bigram language model, removing the bigrams with frequency less than 20 increased the overall accuracy. This is because when we remove the not so frequent bigrams, the frequent bigrams have more say on the prediction.

ii. In the standard language models, it can be observed that fourgrams gave better prediction accuracy than trigrams and trigrams gave better prediction accuracy than bigrams. This is because as the order of the language model increased, the context was captured in an effective manner.

iii. In the forward-backward bigram language model, higher prediction accuracy was obtained when the model was trained on Microsoft n-gram corpus when compared to the case when the model was trained on novels dataset. This is because Microsoft n-gram corpus had larger data resulting in less unseen n-grams.

iv. In the backoff n-gram language model, it can be observed that using weighted sum and non- weighted sum as features has reduced accuracy of predicting the correct answer. This is because, in backoff model, we consider higher order n-gram probabilities and then backoff to lower order n- gram probabilities only if the higher order n-gram probabilities are not available. As the length of the sequence containing the actual option word is more, it is almost certain that the word may occur with same sequence of words. But in models using weighted and non-weighted sum as features, giving weights to lower order n-gram probabilities in the initial stage itself can only increase probabilities of predicting wrong answers.

v. Using only standard trigrams or bigram probabilities as features have much reduced performance in terms of accuracy because they take into account very few n-grams when compared to the backoff n-gram language model.

vi. Inclusion of the root word decreased the accuracy, as the model will look for fourgrams like “**animal has been move**" rather than “**animal has been moving**". Since the probability of the latter occurring is more for grammatical reasons, this model fails in such cases.

7.2 Insights

To gain further insights into our sentence completion model, we looked at how the model was working when the target word is a noun, pronoun, verb, adverb and adjective.

Out of the total 1040 test sentences, 436 had nouns as the target word; 330 of them were correctly filled. 392 had verbs as the target word; 328 of them were correctly filled. 178 had adjectives as the target word; 140 of them were correctly filled. 27 had adverbs as the target word; 22 of them were correctly filled. 7 had pronouns as the target word. All of them were correctly filled.

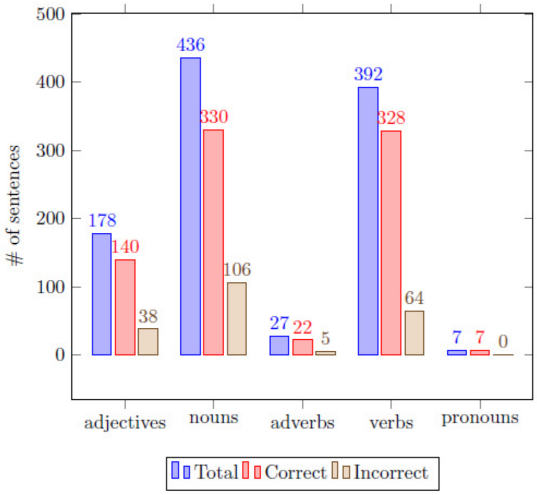


Figure 7.2.1 Histogram for tagged Parts of Speech [Accuracy]

Although the distribution is asymmetrical, from the above Figure 7.2.1, we can see that our model works best in the order - pronouns, verbs, adverbs, adjectives and nouns.

Then we checked how the sentence completion model worked with respect to small and long sentences (word count greater than 8).

From Figure 7.2.2, we can conclude that our model is working better in the case of long sentences than in the case of short sentences. Our fourgram backoff model was able to grasp the context in a good way. The reason for the model not working so well in the case of short sentences is that the sentences didn't have sufficient context for our model.

There are a few sentences in the corpus which have a blank word in between of an idiom. Our model worked very well in completing such sentences.

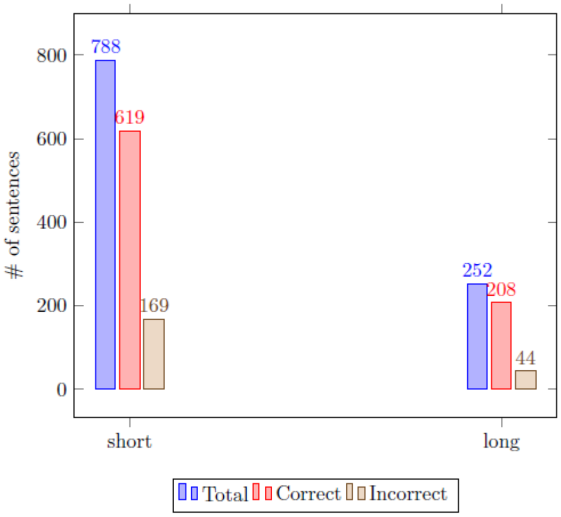


Figure 7.2.2 Histogram for Sentence Lengths [Accuracy]

8. Contributions

Addanki Adithya(aa207): Studied Language Models, preprocessed the data, worked on implementation of Backward and Forward bigram models, and LSA models. Analyzed model accuracy using Microsoft N-gram corpus.

Srinivasa Rao Katta(sk189): Studied Language Models, worked on implementation of Root word based approach to N-Gram model. Analyzed accuracy through Project Gutenberg training set.

9. Future Scope and Conclusion

We have observed various language models, compared the accuracies and analyzed the linguistic features each model offers. We applied the models for deducing the right word in a sentence completion task. The bench mark given by MSR is 92% [1] for the sentence completion task is a distant future for now, which requires further integrations of technology and fields related to computer science are required to imbibe the judgment and intelligence of humans to computers.

10. References

[1] . The MSR Sentence Completion Challenge - http://research.microsoft.com/en-us/projects/scc

[2] . The Microsoft n-gram corpus - http://web-ngram.research.microsoft.com/info/

[3] . Kyusong Lee; Lee, G.G., "Sentence completion task using web-scale data," Big Data and Smart Computing (BIGCOMP), 2014 International Conference on , vol., no., pp.173,176, 15-17 Jan. 2014. doi: 10.1109/BIGCOMP.2014.6741431. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6741431&isnumber=6741395

[4] . Geoffrey Zweig and Christopher J.C. Burges, "The Microsoft Research Sentence Completion Challenge", Dec 2011. http://research.microsoft.com/apps/pubs/default.aspx?id=157031

[5] . Geoffrey Zweig, John C. Platt, Christopher Meek, Christopher J. C. Burges, Ainur Yessenalina, and Qiang Liu. 2012. “Computational approaches to sentence completion”. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1 (ACL '12), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 601-610.

[6] . Bellegarda, J.R., "Exploiting latent semantic information in statistical language modeling," Proceedings of the IEEE , vol.88, no.8, pp.1279,1296, Aug. 2000. doi: 10.1109/5.880084. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=880084&isnumber=19040

[7] . Ahsaee, M.G.; Naghibzadeh, M.; Naieni, S.E.Y., "Weighted Semantic Similarity Assessment Using WordNet," Computer & Information Science (ICCIS), 2012 International Conference on , vol.1, no., pp.66,71, 12-14 June 2012. doi: 10.1109/ICCISci.2012.629721. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6297214&isnumber=6297196