Methods for sentence completion.

Jonatan Ferm
Lund University
Lund, Sweden
pi08jf6@student.lth.se

Abstract

This document describes some basic methods for sentence completion. The methods are first trained on a large corpus of unannotated text, to then try to predict the missing words in the test set which contains just over a thousand sentences where one word is missing and five alternatives for the missing word.

1 Credits

This paper is a result of the course *Language Technology: Project, EDAN60* at the Faculty of Engineering, Lund University. The training- and test set is provided by Microsoft Research via the MSR sentence Completion Challenge (MSR,).

2 Introduction

Sentence completion remains one of the unsolved problems within the domain of natural language processing, with the best results achieving just over 50% correctness (Zweig. et al. 2012,) when given five alternatives. This is probably partially caused by the shifting nature of the problem. Here are two examples from the test set to shine some light on this.

```
Then he did the same with the ____
with which the chamber was
paneled.
a. rowers
b. ceiling
c. wood-work
d. motion
e. speed
```

Here we can see that the correct answer is "woodwork" with which you can panel a chamber. Completing this sentence is in no way trivial and to be certain you are right you have to destruct the sentence into dependencies between the word. If you for example only look at a few words surrounding the missing word it is not at all clear what the correct word is. Another example is the following.

```
The little which I had yet to
learn of the case was told me
by ____ Holmes as we traveled
back next day.
a. Sherlock
b. meeting
c. sending
d. telling
e. permitting
```

In this example I don't think it's as clear what the correct answer should be. If you just look the the next word it seems the answer should be Sherlock. But if you look beyond that, I would argue that you can be told something by meeting someone. Here the correct answer is Sherlock though.

3 The Data Set

This section will briefly cover the data set provided by Microsoft research. Which can be found on their website. (MSR,)

3.1 The Training Set

The training set consists of 524 books containing a total of roughly 41.5 million words. The set is completely unannotated and untokenized, this means that all models in described in this document will suffer from imperfect tokenization.

3.2 The Test Set

The test set consists of 1040 sentences where one word is missing and five alternatives to the missing word. Examples of these sentences can be found in the introduction section of this paper. As is evident the problem is not trivial, current state of the art is just over 50% rate of correctness for machines where as a human reach about 90% (Zweig. et al. 2012,).

4 The Methods

This section will briefly cover the methods used.

4.1 N-Gram

Three models were trained using N-grams, a 1-gram model, a 2-gram model and a 3-gram model. First all the 41.5 million words were tokenized using NLTK (Steven Bird2006,) after which the N-gram probabilities could be calculated. To predict the missing word in a sentence the model calculates

$$P(w_{i-(N+1)} \dots w_{i-1}|w_C) \cdot P(w_C|w_{i+1} \dots w_{i+N-1})$$

$$\forall \{w_C | C \in \text{options}\}.$$

The model uses a simple back-off method when a N-gram is not found reducing it to a (N-1)-gram.

4.2 Logistic Regression

The logistic regression model is implemented using liblinear (Fan et al. 2008,). The training set needed to be processed into a labeled training cases. This was done by first extracting a dictionary of the words in the training set , using only the words that appeared more than ten times, this resulted in about 55000 words. Then iterating over every sentence extracting the words before and after the current word. A training case is then a feature vector with roughly 110 000 input features, consisting of two bags of words one of the words appearing before the current word and one of those appearing after it. The output features is then a vector of 55000. As you may have guessed these feature vectors will be *very* sparse.

5 Results

5.1 N-grams

The result from the N-gram models were as follows.

1-gram	20.4%
2-gram	27.9%
3-gram	37.4%

These models were trained on the full set of 41.5 million words. An attempt to train a 4-gram model was made but the machine it was attempted on did not have enough RAM.

5.2 Logistic Regression

The logistic regression model was trained on a reduced training set containing only the first hundred thousand words. An attempt was made to train a model on the first one million words but after almost two weeks of training there was an error saving the model. The result from training on the first hundred thousand words were a rate of correctness of 22.9%

6 Discussion

6.1 Small data set test

The 3-gram model was also trained on the reduced data set that was used for the logistic regression model. The 3-gram model however only reached a rate of correctness of 19.1%, which is actually worse than simply having a guess at one of the five alternatives. This leads one to think that the logistic model is way more expressive than simply counting the frequencies of 3-grams, since it was actually able to find some patterns in the severely reduced corpus.

6.2 Training time

The increased expressive power of the logistic model comes at a price however. Training the 3-gram model on the full data set takes a matter of minutes which got us a rate of correctness of 37%, whereas training the logistic model on the severely reduced data set took just north of six hours for a result that's marginally better than just guessing. If you consider that the logistic regression model is based of a optimized C-library and the 3-gram model is my own highly unoptimized python model the vast difference in training time gets even more relevant. When you consider the performance improvement in the 3-gram model going from the reduced training set to the full set which is just over 400 times as bigger it becomes clear that it is of vital importance that our models must the able to be trained on large data sets without suffering from too big of a time penalty. Gathering a data set that is several orders of magnitude bigger than our full data set is a trivial task with the vast resources of the internet and then there is no chance for our expressive logistic regression model to compete with models that can actually utilize the full data set.

References

Microsoft Research MSR Sentence Completion Challange http://research.microsoft.com/en-us/projects/scc/

G. Zweig, J. Platt, C. Meek, C. Burges, A. Yessenalina, Q. Liu 2012 Computanional Approaches to Sentence Completion.

Steven Bird 2006 *NLTK: the natural language toolkit*

R. Fan, K, Chang, C. Hsieh, X. Wang, C. Lin 2008 LIBLINEAR: A Library for Large Linear Classification.