UNIVERSITEIT VAN AMSTERDAM

Natural Language Models Interfaces

Enhancing a Hidden Markov Model

Tim Stolp 11848782

March 2019

Contents

1 Introduction

With the ever increasing amount of technology we use, speech recognition has become a big topic. Speech recognition is useful in a whole variety of application such as understanding search engine queries, helping in the health care industry, and automating help desks.

One very effective way to do speech recognition is to use bigrams. Bigrams are models in which you calculate the probability of a sentence by combining the probabilities of each word given its previous word. This method can be effective but it does not fully represent how most languages are actually built. In most languages a sentence structure is made of a combination of parts-of-speech (POS) tags. POS tags are for example nouns, verbs, and adverbs. Hidden Markov Models (HMM) are used to incorporate these mechanics into language models. An HMM works by combining the transition and emission probabilities to predict a sentence. The transition is the probability of a POS tag conditioned by the previous POS tag. The emission is the probability of a word conditioned by a POS tag.

Bigrams and HMMs have different strengths and weaknesses, which brings up the main question if these models can be combined to further improve the overall result of speech recognition. This can be done by interpolating the bigram with the emission and with the transition. To give further insight into how the overall result can be improved these two methods will be researched. In this report a variety of interpolation levels are tried for both the emission and transition. The hypothesis is that the interpolation of the transition will give improvement while the interpolation of the emission will worsen the results because by also conditioning a word with the previous word, instead of just the tag, there would be many more possible outcomes making it more uncertain about what to predict.

2 Method

In python an HMM class is created. This class makes use of the Viterbi algorithm to predict sentences and Laplace smoothing is used to prevent zero probabilities. Laplace smoothing adds the UNK (unknown) tag and the unk word into the database to give unknown words a probability higher than 0. Sentences are made lowercase and padded with BoS (beginning of sentence) and EoS (end of sentence) tags to give the first words a history and to indicate where a sentence ends. The interpolation is done by taking a level of emission and transition interpolation and using the following formulas.

Transition:

$$P_{C|C_{prev}X_{prev}}(c|p,w) = \beta_0 P_{C|C_{prev}}(c|p) + (1-\beta_0)P_{C|X_{prev}}(c|w) \qquad (1)$$

Emission:

$$P_{X|CX_{prev}}(x|c,w) = \beta_1 P_{X|C}(x|c) + (1-\beta_1)P_{X|X_{prev}}(x|w) \qquad (2)$$

Where x is a word, xp is previous word, c is a tag, and cp is previous tag. With the interpolation coefficients $0 \leq \beta_0 < 1$ and $-0 \leq \beta_1 < 1$.

The annotated Penn Treebank Corpus [4] is used for training and testing. Each sentence in this corpus is tagged with the POS tags. This data set consists of 3914 sentences, 3000 of which make up the training set, 100 the validation set, and the remaining 814 the test set. To test the extended HMM a range of 0 to 1 with steps of 0.1 is used for the interpolation coefficients of the emission and transition separately.

3 Results

The interpolation levels result in a total of 100 different accuracies. The resulting accuracies are cut off to 3 decimals for readability and put into a table (table 1). The results are also plotted in a 3d graph (figure 1) to give a clear view on the effects of different amounts of interpolation.

The highest accuracy achieved was with an emission interpolation coefficient of 0.0 and a transition interpolation coefficient of 0.4. The highest average accuracy for transition interpolation is with a coefficient of 0.6 and the highest average accuracy for emission interpolation is with a coefficient of 0.0.

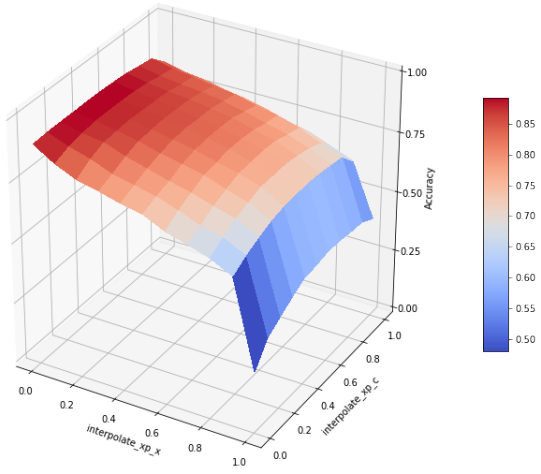

Figure 1: Accuracies of different amount of interpolation

Transition interpolation coefficient	Emission interpolation coefficient											
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	Average
0.0	0.894	0.840	0.801	0.781	0.758	0.740	0.698	0.673	0.645	0.601	0.236	0.697
0.1	0.898	0.864	0.839	0.803	0.785	0.764	0.735	0.699	0.678	0.635	0.326	0.730
0.2	0.899	0.876	0.863	0.831	0.807	0.782	0.761	0.733	0.708	0.661	0.362	0.753
0.3	0.899	0.883	0.868	0.850	0.826	0.807	0.788	0.755	0.728	0.683	0.397	0.771
0.4	**0.899**	0.883	0.872	0.855	0.838	0.821	0.796	0.776	0.747	0.695	0.429	0.783
0.5	0.899	0.883	0.869	0.858	0.843	0.827	0.806	0.779	0.748	0.699	0.434	0.786
0.6	0.896	0.880	0.867	0.856	0.839	0.826	0.806	0.783	0.749	0.702	0.445	**0.786**
0.7	0.891	0.874	0.861	0.852	0.839	0.824	0.799	0.778	0.742	0.698	0.443	0.782
0.8	0.881	0.862	0.850	0.843	0.827	0.809	0.785	0.762	0.729	0.690	0.435	0.770
0.9	0.872	0.854	0.843	0.827	0.816	0.794	0.776	0.748	0.709	0.680	0.430	0.759
1.0	0.819	0.800	0.788	0.770	0.755	0.734	0.709	0.689	0.653	0.618	0.402	0.703
Average	**0.886**	0.864	0.847	0.830	0.812	0.793	0.769	0.743	0.712	0.669	0.394	

Table 1: Accuracies of different amount of interpolation

4 Discussion

From the results the first thing to notice is that using emission interpolation has a clear negative effect. The higher the emission interpolation coefficient the lower the resulting accuracy. (Figure 1) This is in line with the expected results. By also looking at the previous word when predicting the next word one would expect that this would allow for many more possible words to choose from, making it less clear which word to predict. On the other hand transition interpolation seems to have a slight positive effect maxing out around a coefficient of 0.6 and decreasing when nearing 0 and 1. This is also in line with the expected results.

There are multiple things that require further research. First of all while this report only goes into using bigrams, it is also possible to use other n-grams to combine with the HMM. This might be beneficial since trigrams already have a higher accuracy when predicting sentences [3]. Indicating that using trigrams would likely give a better result. Secondly, other smoothing techniques can be used. Katz smoothing and Jelinek-Mercer smoothing tend to perform well when using bigrams and trigrams. [1] Lastly, combining n-grams and HMMs with other language models such as neural networks [2] might further improve the results. This could be done in a variety of different combinations and might give further insight into the advantages and disadvantages of each model.

5 Conclusion

When combining a Hidden Markov Model with a bigram on both the emission and transition it has an overall negative effect on the performance of the language model. The biggest effect is seen when applying the interpolation to the emission. This has a big negative effect on the performance. When applying the interpolation to the transition there is a small positive effect. All in all it is possible to improve the performance of a Hidden Markov Model by combining part of it with a bigram.

References

[1] Stanley F. Chen and Joshua Goodman. "An Empirical Study of Smoothing Techniques for Language Modeling". In: (), pp. 310–318.

[2] Kejin Jin. *Language Model: A Survey of the State-of-the-Art Technology.* `https://medium.com/syncedreview/language-model-a-survey-of-the-state-of-the-art-technology-64d1a2e5a466`. 2017.

[3] Robert Lewand. "Cryptological Mathematics. The Mathematical Association of America." In: s1 (2000), p. 37.

[4] *NLTK Data.* `http://www.nltk.org/nltk_data/`. 44. Penn Treebank.