A Survey of Predictive Text

Joshua Leppo[1]

[1]Computer Science, Penn State Harrisburg, Penn State University, 777 West Harrisburg Pike, Middletown, 17057, PA, USA.

Abstract

TODO

Keywords: TODO

1 Introduction

Natural Language Processing (NLP) is a field within machine learning that deals with human language. While there are many subfields within NLP, in this literature survey, we will review text prediction within the text processing field. A computer's ability to recognize text patterns and make meaningful and useful predictions about which characters or words might come next has a variety of practical applications.

Text predictions can be used to speed up the time it takes to respond to emails or write letters. It can lead to decreased spelling and grammatical errors. It can assist with the translation from one language to another. These abilities can benefit business by increasing productivity and efficiency. Individuals with cognitive or motor impairments will benefit from needing less keystrokes to complete their writing as well as minimizing errors. There are a number of practical applications that have an even greater number of benefits. Research on NLP text prediction methods is required to ensure efficient, useful text predictions.

In this literature survey, we will explore twelve papers ranging from 2005 up to and including 2023. There are three papers from 2020 to current, three papers from 2015 to 2019, three papers from 2010 to 2014 and three papers written before 2010. This should provide a range of views and methods on the topic of text prediction, covering almost twenty years of research.

2 Papers Before 2010

In this section, we review [1], which predicts sentence completion using an n-gram model with beam search. [2] focuses on single word predictions and a user interface for efficient use of the model. In [3], word disambiguation is approached using a classification system rather than a straight forward prediction model.

2.1 Predicting Sentences using N-Gram Language Models

In [1], the authors address the problem of text prediction using an N-Gram language model. The Viterbi principle is used to simplify the problem and improve efficiency. The model attempts to address the issue of predicting subsequent words given some beginning sentence fragment. This feature has the ability to greatly increase productivity in menial repetitive tasks, such as answering emails or entering command line commands in a terminal.

Since the purpose is to improve productivity, the model quantifies improvement by considering only perfect predictions. Consider a text prediction for an email draft. If the user accepts the prediction, but still needs to go back to make corrections, then the prediction did not improve efficiency. For this reason, the results only consider a good prediction one in which the prediction was accepted in its entirety and not corrected.

They use an instance-based method of text prediction as their base line and compare that to the performance of their model. They use a diverse corpus of emails, weather reports and recipes. Evaluation is based primarily on two components. They look at precision, which is the ratio of accepted predictions, and they look at recall, which is a ratio that essentially quantifies the number of keystrokes saved by the accepted prediction. Ultimately, their research shows increased precision and recall where training documents have less diversity. For example, service center emails achieved much higher keystroke savings than recipes. However, in all situations, their model provides some benefit in improved efficiency and productivity.

2.2 Advances in NLP applied to Word Prediction

The approach in [2] focuses on individual word prediction rather than sentence completion. Word prediction greatly benefits individuals with cognitive or motor disabilities by greatly reducing the amount of time it takes to write and reducing the amount of potential errors due to misspellings. They developed a method that includes both statistical analysis as well as rule-based features called FastType.

FastType is a user interface designed to decrease necessary keystrokes while improving spelling. The authors improve on the original FastType by implementing DonKey, which is simply an improved interface with an improved underlying language model to improve predictions. Predictions are improved by adding POS n-gram models and Tagged Word n-gram models together in

their model. The DonKey user interface simply provides a ranked list of suggestions based on the typed characters. The user can choose a word from the list by clicking a button that corresponds to the word or by using a mouse or other pointing device.

The system is evaluated using three metrics, the ratio of keystrokes saved, the number of keystrokes until the correct word appears in the list and the amount of time that is saved by using the system instead of typing the full word out. They tested using list lengths of 5, 10 and 20. While there was some variation between the list length, the keystrokes saved ratio was between about 46% and 55%, increasing with list length. Keystrokes until completion ranged from 2.55 to 2.06, decreasing with list length. The amount of time saved was about 25% to 29%, increasing with list length. In all areas, their improvements to the language model and their improvements to the user interface produced improved results when compared with current benchmarks at their time.

2.3 A Learning-Classification Based Approach for Word Prediction

In this paper, [3] looks at word prediction as a classification problem. They use machine learning methods with feature extraction techniques that are developed from Mutual Information and Chi-Square techniques. This model is able to produce candidate words from a given context and then classify them to predict the most likely next word.

Given a corpus text, multiple sets of "confusion" words, or words that are often confused, are extracted along with some number of words preceding them. The features they extract are not just the preceding words but are words that are more helpful in telling one confused word from another, which is largely what sets their model apart from other similar feature extraction language models. These features are selected using Mutual Information and Chi-Square techniques and are arranged into feature vectors.

The vectors are used to train the classifier using Support Vector Machines (SVM). Each confused word is used as a class to train the classifier. This is done for each set of confused words, so that they each have a classifier model. Predictions are made by considering the given context and classifying it to the most likely confused word.

The evaluation of the model was done using Naïve Bayes as the baseline and a number of different datasets taken from various websites and databases. They used three confusion sets comprised of various confused words used extensively in word prediction research. The Bayesian approach outperformed their method in only one of the datasets and only by about 1%. The other three datasets saw increases over the Bayesian approach by as much as 7%. Overall, their approach is able to produce high accuracy while considering only small amounts of context.

3 Papers Between 2010 and 2014

TODO

4 Papers Between 2015 and 2019

TODO

5 Papers Since 2020

TODO

6 Conclusion

TODO

References

[1] Bickel, S., Haider, P., Scheffer, T.: Predicting sentences using n-gram language models. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 193–200 (2005)

[2] Aliprandi, C., Carmignani, N., Deha, N., Mancarella, P., Rubino, M.: Advances in nlp applied to word prediction. University of Pisa, Italy February (2008)

[3] Al-Mubaid, H.: A learning-classification based approach for word prediction. Int. Arab J. Inf. Technol. **4**(3), 264–271 (2007)