Context based word prediction for Mobile devices with T9 keypads

Apoorva Saxena

Abstract—The use of digital mobile phones has led to a tremendous increase in communication using messaging. On a phone T9 keypad, multiple words are mapped to same numeric code due to multiple posssilbe combinationsof different alphabets associated with the particular numeric code . The article proposes a Context Based Word Prediction system for SMS messaging in which context of the word based on prior words is used to predict the most appropriate word for a given code.

Index Terms— Context-based Word Prediction Algorithm,SMS messaging ,T9 keypad

1 INTRODUCTION

The growth of wireless technology has provided us with many new ways of communication such as SMS (Short Message Service). SMS messaging can also be used to interact with automated systems or participating in contests. With tremendous increase in Mobile Text Messaging, there is a need for an efficient text input system. With limited keys on the mobile phone, multiple letters are mapped to same number (8 keys, 2 to 9, for 26 alphabets). The many to one mapping of alphabets to numbers gives us same numeric code for multiple words.

Predictive text systems in place use the frequency-based disambiguation method and predict the most commonly used word above other possible words. T-9 (Text on 9-keys), developed by Tegic Communications, is one such predictive text technology used by LG, Siemens, Nokia Sony Ericson and others in their phones. T-9 system predicts the correct word for a given numeric code based on frequency. This may not give us the correct result most of the time. For example, for code '63', two possible words are 'me' and 'of'. Based on a frequency list where 'of' is more likely than 'me', T-9 system will always predict 'of' for code '63'. So, for a sentence like 'Give me a box of chocolate', the prediction would be 'Give of a box of chocolate'.

The sentence itself indeed gives us information about what should be the correct word for a given code. Consider the above sentence with blanks, "Give _ a box _ chocolate". According to the English grammar, it is more likely that 'of' comes after a noun 'box' than 'me' i.e. it is more likely to see the phrase "box of" than "box me". The algorithm proposed is an online method that uses this knowledge to correctly predict the word for a given code considering its previous context. In the proposed method, the context information is used to choose the appropriate word.

2 PROPOSED METHOD

2.1 Workflow

The proposed method uses machine learning algorithms to predict the current word given its code and previous word's Part of Speech (POS). The workflow of the system is as shown in Figure 1. The algorithm predicts the current word after training a Markov Model on Enron email corpus since short emails resemble SMS messages closely.

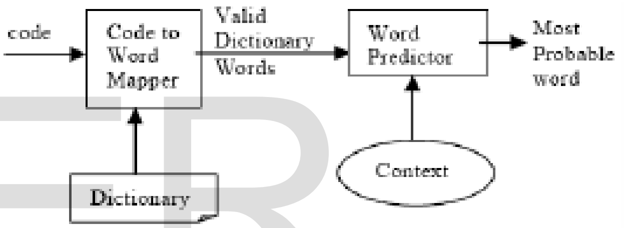

Figure 1: Workflow for Context Based Word Prediction System for formal lanaguge

The code, word and its POS are three random variables in the model. The dependency relationship between these variables can be modeled in different ways and we analyse and present a discussion of pros and cons of each modeling approach. The appropriate modeling of a given problem is a design issue and we present our detailed design approach in this paper for the given problem at hand. The first-order markov model with different representations of this relationship is discussed below. The bi-gram language model (Manning and Schütze, 1999) is used to predict the most probable word and POS pair given its code and previous word's POS.

2.2 Markov Model-I

In this first order Markov model (Figure 2), word is dependent on its code and the part of speech is dependent on the word and part of speech of previous word. Here, in a sentence, C_t refers to the numeric code for t^{th} word, W_t refers to t^{th} word and S_t refers to the part-of-speech of t^{th} word. Let W_{t+1} W_t be a sequence of words where W_{t+1} is to be predicted and W_t is known. Also, C_{t+1} and S_t are known.:

$$P(W_{t+1}S_{t+1}/C_{t+1}S_t) = P(W_{t+1}C_{t+1}S_{t+1}S)/P(C_{t+1}S_t) \quad .. (1)$$

IJSER

The joint probability distribution using factorization theorem is given as,

$P(W_{t+1}C_{t+1}S_{t+1}S_t)=P(S_{t+1} / W_{t+1}S_t)P(W_{t+1} / C_{t+1})P(C_{t+1})P(S_t)$ ..(2)

Hence,

$P(W_{t+1}\, S_{t+1}/C_{t+1}S_t) =$

$P(S_{t+1}/W_{t+1}S_t)P(W_{t+1}/C_{t+1})P(C_{t+1})P(S_t) \quad / \quad P(C_{t+1}S_t)$ ..(3)

Where,

$P(C_{t+1}S_t)=$

$\sum_{W_{t+1},S_{t+1}} \quad P(W_{t+1}C_{t+1}S_{t+1}S_t)$ ..(4)

$(W_{t+1}S_{t+1})= \arg\max_{W_{t+1},S_{t+1}} P(W_{t+1}S_{t+1} / C_{t+1}S_t)$

The word for which the above joint probability (word and its part of speech) is highest given its numeric code and previous word's part of speech is chosen. In order to predict first word of the sentence, we assume a null word preceding it, which denotes the beginning of the sentence. The null word also represents the context of the word as not every word can start a sentence.

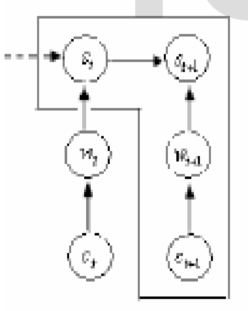

Figure 2: Markov Model-I for Context based word prediction

2.3 Markov Model-II

Here the code is dependent on its corresponding word and the word is dependent on its part of speech. This appears to be a more intuitive way of expressing the relationship from the user's perspective as when the user enters a code; he/she has the word in mind and not the code. The POS of consecutive words have a causal relationship which encodes the grammar of the sentence.

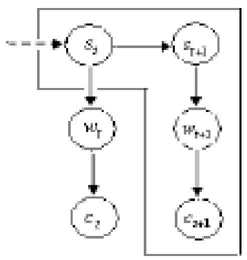

Figure 3: Markov Model-II for Context based word prediction

The joint probability distribution using factorization theorem is given as,

$P(W_{t+1}C_{t+1}S_{t+1}S_t)=P(S_{t+1} / S_t)P(W_{t+1} / S_{t+1})P(C_{t+1}/W_{t+1})P(S_t)$ ..(5)

Hence,

$P(W_{t+1}\, S_{t+1}/C_{t+1}S_t) =$

$P(S_{t+1}/S_t)P(W_{t+1}/S_{t+1})P(C_{t+1}/W_{t+1})P(S_t) \quad / \quad P(C_{t+1}S_t)$ ..(6)

Where,

$P(C_{t+1}S_t)=$

$\sum_{W_{t+1},S_{t+1}} \quad P(W_{t+1}C_{t+1}S_{t+1}S_t)$ ..(7)

$(W_{t+1}S_{t+1})= \arg\max_{W_{t+1},S_{t+1}} P(W_{t+1}S_{t+1} / C_{t+1}S_t)$

IJSER

2.4 Support Vector Machines (SVMs)

SVM has been used in sequence tagging for predicting the POS sequence for a given word sequence. Hidden Markov Support Vector Machines uses a combination of SVM and Hidden Markov Model for sequence tagging. SVM^{HMM} is implemented as a specialization of the SVMstruct package for sequence tagging.

In the given problem, the correct word is to be predicted. using SVM for this purpose would require as many classes as number of words in the dictionary. The English dictionary has roughly around 100,000 words; SVM would need to learn classification for these many classes. To learn a good SVM classifier for 100,000 classes, sufficiently large number of examples is required for all the classes i.e. a large training dataset which covers words from all these classes but the training time for SVM grows exponentially with the number of training examples.

However, for the given problem of predicting the correct word for a given code, one classifier per code is really what we need to learn. But the number of codes can be very large as well (# of digits in code = #of letters in word). Hence, to use SVM for this problem, the number of codes needs to be limited. The features used for SVM are similar to parameters used in the above graphical models i.e. the POS tag of previous word and the given code. SVMHMM was used for implementation.

3 Analysis

3.1 Experiment

The training was done on about 19,000 emails (Enron Email Data) and the testing was done on about 1900 emails, with each email consisting of 300 words on average. The English dictionary available on Linux system was used. Results were compared with frequency based estimation method using the frequency list from Wikipedia. The results are documented in Table 1.

Training Examples	Test Examples	Avg % error in Markov Model		Frequency Based Model
		Markov Model I	Markov Model II	
19000	1900	5.5%	6.7%	8%

Table 1: Test Results for Context Based Word Predicton System for formal language

3.2 Analysis of Markov Models

In Markov Model-II and Markov Model-III, the Part Of Speech (POS) of the current word is determined only by the POS of the previous word. However, the current word also plays an important role in determining the POS. As observed in the training data and is intuitive as well, the POS ‘IN’ (preposition) is more likely to have a POS ‘CD’ (Cardinal number) following it than a ‘PRP’ (Personal pronoun). E.g. CD follows IN – “About 20% increase in sales was observed this year” and PRP following IN – “They were concerned about me”. But given a code “63”, which maps to the number “63”and word “me”, it is more likely that “me” comes after a preposition (like about) than a number “63”. Thus, current word and previous POS together determine the current POS. This is modeled in Markov Model-I and Markov Model-IV.

In models Markov Model-II and Markov Model-IV, the word determines the code. However, given the word, code is deterministic i.e. there is only one possible code for a given word. But given a code, word corresponding to it is determined probabilistically based on the context. Also, for the predictive system, code is known and we need to find the most probable word for it. Thus, Markov Model-II and Markov Model -IV do not model the causal relationship between word and code appropriately and hence they perform worse than Markov Model -I.

Given all the analysis above, Markov Model-I models the given problem the best and as also observed it gives the best performance. Our analysis of different ways of modeling the problem shows that it’s very important that the causal relationship between different variables is modeled correctly to develop an efficient system. And this analysis may require the domain knowledge of the problem at hand.

3.3 Performance of SVM

To assess how SVM performs in classifying the words for a given code, it was tested on 4 codes corresponding to a few very frequent English words. Comparison of SVM, graphical model and frequency method on these words is shown in Table 2. SVM performs better than frequency method and reduces the average error (% of words incorrectly identified) by 10% approx. However, Markov Model-I outperforms SVM by reducing the error further by 30% approx. Markov model performs better than SVM because causal relationships between variables can be better modeled in a Markov model.

Words	SVM	Markov Model I	Frequency Based Method
63: (of,me)	25.4%	25.5%	25.0%
43: (he,if)	24%	28%	76.4%
84373: (there,these)	52%	46%	49%
66: (on,no)	89%	11%	80%
Average Error	47.6%	27.6%	57.6%

Table 2: Test Results for comparison of graphical model and SVM

3.4 Conclusion and Future Work

The Context Based Word Prediction system performs better than the traditional frequency based method. Different Markov models were analyzed to judge what best models the causal relationship between parameters. SVM^{HMM} model used for sequence tagging was found to be inappropriate for the given problem due to the large number of classes. The bi-gram model used can be extended to tri-gram or more but since SMS text messages are normally short sentences, a higher N-gram model wouldn’t be useful.. Currently, we model the first order Markov dependency only between the POS of the consecutive words. Modeling this dependency among the consecutive words themselves might give an improvement in performance because certain word bi-grams are more likely than others. This might be a good extension to the current system. In the current model, error made at the tth word is propagated further in the sequence and hence the error for the current word also reflects the error made on the previous word (on the basis of which it was predicted). However, in a mobile messaging system, a user can actually correct the word if the word proposed by the system is wrong and hence it would be better to predict the current word based on the actual (correct) previous word. For unseen words, a very low probability is assigned to them and probabilities of all the words for a given code are normalized

IJSER

ACKNOWLEDGMENT

I am very grateful towards my project guide Prof. T J Siddiqui whose precious guidance helped me with my research..

REFERENCES

[1] Altun, Y., Tsochantaridis, I., & Hofmann, T. (2003). Hidden Markov Support Vector Machines. In Proceedings of 20th International Conference on Machine Learning.

[2] Rabiner, L. R., (1989) A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition

[3] Context Based Word Prediction for Texting1 Language, Sachin Agarwal & Shilpa Arora, Carnegie Mellon University

[4] T-9 System. http://www.t9.com/

[5] Enron Email Data. http://www.cs.cmu.edu/%7Eeinat/datasets.html

[6] SVM^{HMM}. http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html

[7] Wikipedia Frequency List. http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists

IJSER