Context Based Word Prediction for Texting[1] Language

Sachin Agarwal
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
sachina@cs.cmu.edu

Shilpa Arora
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
shilpaa@cs.cmu.edu

Abstract

The use of digital mobile phones has led to a tremendous increase in communication using SMS. On a phone keypad, multiple words are mapped to same numeric code. We propose a Context Based Word Prediction system for SMS messaging in which context is used to predict the most appropriate word for a given code. We extend this system to allow informal words (short forms for proper English words). The mapping from informal word to its proper English words is done using Double Metaphone Encoding based on their phonetic similarity. The results show good improvement over the traditional frequency based word estimation.

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, such as ordering products and services for mobile phones, 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 words above other possible words. T-9 (Text on 9-keys)[1], developed by Tegic Communications, is one such predictive text technology used by LG, Siemens, Nokia Sony Ericson and others in their phones. iTap is another similar system developed and used by Motorola 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

[1] SMS Text language

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.

An extension of T-9 system called T-12 was proposed by UzZaman et. al [2]. They extend the idea of T-9 to what we call informal language which is used in Text messaging a lot. This includes abbreviation, acronyms, short forms of the words based on phonetic similarity (e.g. gr8 for great). They use the Metaphone Encoding [3] technique to find phonetically similar words. And from among those phonetically similar words, they choose the appropriate word using string matching algorithms such as edit distance between the word and its normalized form. However, the edit distance measure suggests words such as ‘create’ for informal word ‘gr8’. In the proposed method, the context information is used to choose the appropriate word.

2 Problem Statement

The current systems for word prediction in Text Messaging predict the word for a code based on its frequency obtained from a huge corpus. However, the word at a particular position in a sentence depends on its context and this intuition motivated us to use Machine Learning algorithms to predict a word, based on its context. The system also takes into consideration the proper English words for the codes corresponding to the words in informal language.

Although the method has been proposed for a text messaging system, it is applicable in a number of other domains as well. The informal and formal language mixture discussed here is also used in instant messaging and emails. The proposed method can also be used to convert a group of documents in informal language into formal language. These days, even (non-personal) discussions over emails/IM between friends, colleagues, students is done in a more informal language but if someone were to make use of these discussions formally, then our system can automatically do the conversion or suggest appropriate conversions.

3 Proposed Method

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 using Enron email corpus [4] since short emails resemble SMS messages closely.

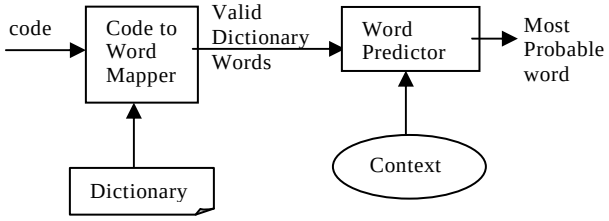


Figure 1: Workflow for Context Based Word Prediction System for formal lanaguge

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. The graphical models with different representations of this relationship are discussed below. The bi-gram model is used to predict the most probable word and POS pair given its code and previous word’s POS.

I. First Model

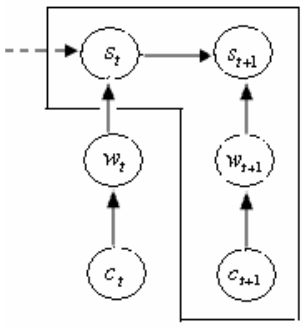


Figure 2: Graphical Model used for Context based word prediction

In this model, word is dependent on its code and the part of speech is dependent on the word and part of speech of previous word. Here, C_t refers to the numeric code for t^{th} word in a sentence. W_t refers to t^{th} word in a sentence and S_t refers to the part-of-speech of t^{th} word in the sentence. 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. We need to learn the $P(W_{t+1}S_{t+1}/C_{t+1}S_t) = \frac{P(W_{t+1}C_{t+1}S_{t+1}S_t)}{P(C_{t+1}S_t)}$

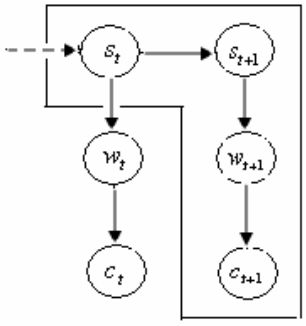
The joint probability distribution from the graphical model using factorization theorem is given as, $P(W_{t+1}C_{t+1}S_{t+1}S_t) = P(S_{t+1}/W_{t+1}S_t)P(W_{t+1}/C_{t+1})P(C_{t+1})P(S_t)$

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

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

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

II. Second Model



HMM-I

Figure 3: Graphical Model used for Context based word prediction(using HMMs)

In this model, a Hidden Markov model is used. The part of speech is the hidden variable and codes are the emissions. 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.

The joint Probability distribution as calculated from the graphical model using factorization theorem is given as

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

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

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

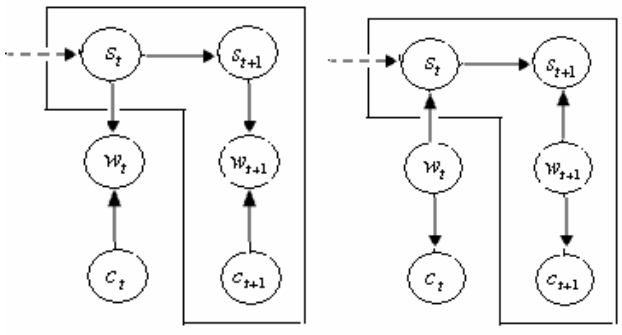
In both the models, the word for which the above joint probability (word and its part of speech) is highest given its numeric code and previous word's part of speech is chosen. In order to predict first word of the sentence, we assume a null word preceding it, which denotes the beginning of the sentence.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

Variations of Model 2:

The HMM model above may be counter-intuitive because of the following reasons:

1. In the model, part of speech determines the word; where as normally the word determines the part of speech. A variation of the above model in which the dependency between current word and its part of speech is reversed is as shown in figure 4 (HMM-III).
2. In the model, word determines the code but for the prediction system, code is given and that determines the possible words, so looking from the prediction system's perspective, it is more intuitive to have the code determine the possible word. This is depicted in the second model in figure 4 (HMM-II).

The first HMM model will be referred to as HMM-I and the two variations as HMM-II and HMM-III.



HMM-II **HMM-III**

Figure 4: Variations in HMM model for Context Based Word Prediction

For HMM-II., the joint Probability distribution from the graphical model would be

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

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

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

For HMM-III., the joint Probability distribution from the graphical model would be

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

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

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

III. 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 [5] uses a combination of SVM and Hidden Markov Model for sequence tagging. SVM^{HMM} [6] is implemented as a specialization of the SVM^{struct} [7] 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 100,000 words but even with a smaller dictionary of say 20,000, SVM needs to learn classification for these many classes. To learn a good SVM classifier for 20,000 classes, sufficiently large number of examples is required for all the classes i.e. a large training dataset which covers words from all the classes.

However, for the given problem of predicting the correct word for a given code, one classifier per code needs to be learnt. 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. SVM^{HMM} was used for implementation.

IV. Informal Language Model

The workflow for the Informal Language Model is as follows:

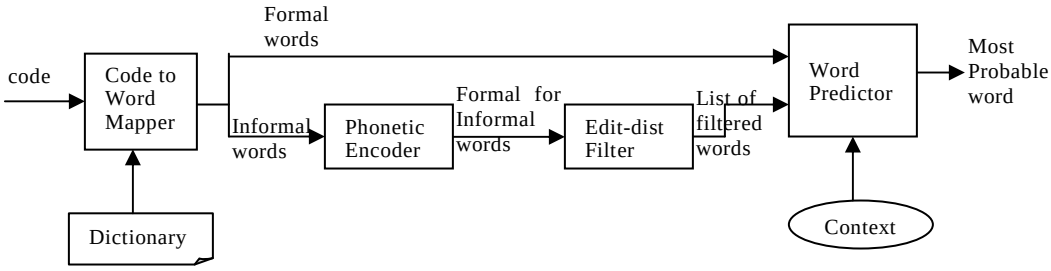


Figure 5: Workflow for Context Based Word Prediction System for Informal lanaguge

The input code is processed to generate all possible words corresponding to that code. These words are split into formal and informal words based on the dictionary look up. The phonetically similar formal words are generated for the informal words using Double Metaphone [8] encoding. This gives a big list of possible formal words. The words in this list are filtered out based on their edit distance from the informal word. Levenshtein Distance [9] is used as the edit distance measure. The final set of possible words for the given code includes its formal words and this filtered list. The trained model for the formal language discussed above is then used to predict the most probable word from this list. Preference is given to the formal words over this list of filtered words as normally the user would enter formal text with some commonly used informal words.

Even after filtering the words based on edit distance, the list still contains the words which are not commonly used in the Texting language. Threshold of two is used for edit distance. If the threshold is reduced further, some of the legitimate formal words are filtered out. For example, edit distance between ‘you’ and its commonly used short form ‘u’ is 2. Hence, reducing threshold further would filter out ‘you’. Therefore, an encoding scheme with better precision in the given domain is required. With the given encoding scheme, a lookup list was used to filter out the informal words that are not

commonly used in Texting language. This lookup list was generated from the SMS message corpus used and a lexicon of SMS available online called "MobiLingo" [10].

NOTE: The informal words considered here are phonetic short forms of a formal word. We do not handle short forms that are abbreviation of a set of formal words, e.g. 'lol' is commonly used short form for 'laugh out loud', but they are not phonetically similar.

4 Experiments

The training was done on about 19,000 emails and the testing was done on about 1900 emails, with each email consisting 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 [11]. The results are documented in Table 1. As can be seen the error reduces by approximately 31% for the first model and 16.8% for HMM-I model. The error for HMM-III model is similar to frequency based method and for HMM-II model the error is more than frequency based.

# Training examples	# Test examples	Avg. % error Model-I	Avg. % error HMM			Avg. % error Freq. based prediction
			I	II	III	
19000	1900	5.54%	6.69%	11.97%	8.05%	8.04%

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

4.1 Analysis of HMM models

Analysis 1: In HMM-I and HMM-II, 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 our 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 an IN – "About 20% increase in sales was observed this year" and PRP following an 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 "63". Thus, current word and previous POS together determine the current POS. This is modeled in Model-I and HMM-III.

Analysis 2: HMM-II has a V-structure between current POS, word and code (Figure 6). According to this model, given the word, code and POS become dependent. However, in the given problem, once we know the word, code doesn't give us any additional information about POS i.e. POS becomes independent of the code given the word. And if the word is not observed, knowing the code increases the probability of POS tags corresponding to the words of that code. Thus, the causal relationship between code and POS is not modeled correctly in HMM-II and hence it performs worse than other models.

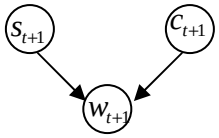


Figure 6: V-structure relationship for code, word and POS

Analysis 3: In models HMM-I and HMM-III, 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 our predictive system, code is known and we need to find the most probable word for it. Thus, HMM-I and HMM-III do not model the causal

relationship between word and code appropriately and hence they perform worse than Model-I.

Given all the three analysis above, Model-I models the given problem the best and as also observed it gives the best performance.

Note: While calculating the error, we ignore the special characters and non-dictionary words. Also, we do not handle the date-time format, email address and hyperlinks. Non-dictionary words like Proper nouns, email addresses and hyperlinks can be unlimited and hence would not be handled by our system as is the case with the current mobile text messaging systems.

4.2. SVM Testing

To assess how SVM performs in classifying the words for a given code, it was tested on 10 codes corresponding to 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 by 18.62%. However, Model-I outperforms SVM by reducing the error further by 35.75%. Graphical model perform better than SVM because causal relationships between variables can be better modeled in graphical models.

Words	**SVM** (c^2=0.1,e^3=0.5)	**Model I**	**Frequency Method**
63: (of, me)	27.43%	27.48%	27.48%
46: (in,go,io)	4.74%	5%	5%
43: (he,if)	24.99%	29.2%	76.42%
64448: (night,might)	36.84%	37.59%	63.15%
843: (the,tie,vie)	0.08%	0.08%	0.08%
84373: (there,these)	51.74%	46.95%	49.12%
8436: (then,them)	63.68%	18.24%	63.68%
66: (no,on)	90.21%	11.94%	88.89%
4283: (have, hate, gave,gate)	1.58%	1.57%	1.57%
87: (up,us)	33.11%	36.76%	35.57%
Average Error	**33.44%**	**21.48%**	**41.10%**

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

4.3 Informal Model

For testing Informal Model, data set of about 850 SMS messages with informal language from [12] was used. Model-I and three models for HMM were tested on the informal data and their results were compared with frequency based model. Results are as shown in Table 3.

# Training examples	**# Test examples**	**Avg. % error Model-I**	**Avg. % error HMM**			**Avg. % error Freq. based prediction**
			I	**II**	**III**	
19000	850	25.24%	25.94%	29.75%	26.88%	33.4%

Table 3: Test Results for Context Based Word Predicton System for formal lanaguge

Model-I performs the best for informal language and it reduces the error by 22.33% over the frequency based method.

[2] Weight for slack term

[3] Precision to which constraints are required to be satisfied by the solution

5 Conclusion

The Context Based Word Prediction system performs better than the traditional frequency based method. Different graphical 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 gram model wouldn't be useful. Phonetic encoding scheme with more precision in the given domain would help improve performance of the Informal Model.

6 Acknowledgements

We are thankful to Tom Mitchell and Eric Xing whose precious guidance helped us in proceeding with our ideas to their actual implementation. We are very thankful to Yifen Huang who helped us by clearing our doubts at various stages of the project.

7 References

[1]. http://www.t9.com/

[2]. UzZaman, N., Khan, M. "T12: An Advanced Text Input System with Phonetic Support for Mobile Devices", 2nd International Conference on Mobile Technology, Applications and Systems, p.1-7(2005).

[3]. Lawrence Philip's Metaphone Algorithm (http://aspell.net/metaphone/)

[4]. http://www.cs.cmu.edu/%7Eeinat/datasets.html

[5]. Altun, Y., Tsochantaridis, I., & Hofmann, T. (2003). Hidden Markov support vector machines. Proc. ICML.

[6]. http://www.cs.cornell.edu/People/tj/svm_light/svm_hmm.html

[7]. http://svmlight.joachims.org/

[8]. http://en.wikipedia.org/wiki/Double_Metaphone

[9]. http://www.merriampark.com/ld.htm

[10]. http://www.mobimarketing.com/downloads/mlingo.pdf

[11]. http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists

[12]. http://www.mla.iitkgp.ernet.in/~monojit/sms.html