Personalized Language Models for Computer-mediated Communication

Umme Hafsa Billah[1], Sheikh Muhammad Sarwar[2], Abdullah-Al-Mamun[3]

[1] Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh
hafsabillah@yahoo.com
[2] Institute of Information Technology, University of Dhaka, Dhaka, Bangladesh
smsarwar@du.ac.bd
[3] Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh
mamun05@uap-bd.edu

Abstract. In this paper, we investigate the performance of statistical language models on Instant Messaging (IM) data. Language Models (LM) are quite useful for modeling text data, and hence they are helpful in different contexts like *spelling correction*, *speech recognition*, *part-of-speech tagging* etc. Construction of LM on a users past messaging data would be a strategy to model her writing style, and that LM can then be used to predict the next word in her future communications. However, we hypothesize that a user follows a specific pattern of communication with each of her virtual acquaintances. As a consequence, LM built on her entire messaging history would degrade the performance of the next word predictor, while communicating with a specific person. In this paper, we deploy a special method that excludes some specific message contents from the entire history in order to build LM. Our method suggests that, at the time of communicating with a specific user, a special LM should be invoked from a set of models for increasing accuracy. We analyze the IM data of a set of users, and show that our method performs well in terms of perplexity.

Keywords: Language Model (LM), Perplexity

1 Introduction

People have conversation with each other almost every day using computing systems as a media; the applications or software used for this purpose are usually referred to as Instant Messaging (IM) system. It has become one of the mostly used paradigms for communication. As a result, it is integrated as a service with different types of social networks and e-mailing systems. As for example, Facebook and Gmail provide instant messaging facility as a part of their core services. People who use these systems, often need to communicate with friends, relatives, business collaborators etc. Generally, people use a certain way of typing, while

having casual conversation with another person. In order to facilitate and expedite such personalized typing, most IM software includes a component, that predicts and suggests a set of words, given the current input words of the message sender. Thus the next word prediction component is beneficial for everyday, as it reduces the time consumed for typing.

Human communication is predominantly personalized in real world *i.e.* a person internally uses and manages a specific dictionary for finding appropriate words to chat with another specific person. As for example, people exchange messages with their work groups formally using phrases like *With reference to our conversation previously, Yours sincerely* etc. On the other hand, at the time of exchanging messages with family or friends, they use casual phrases like *Hey wassup!, How you doing?* etc. Languages like Bengali are exposed to more personalized level of communication. For example, there are three different words for addressing a person: *tumi*, *tui* and *apni*, in Bengali against a single word *you* in English. Based on these three types of addressing, for a single sentence in English, there can be three possible sentences in Bengali with the same meaning. Some samples are shown in table 1. Thus, a personalized prediction system would be a contribution to gear up the typing speed while having informal computer based communication; especially for the case of languages like Bengali. As a result, the need for constructing personalized language based statistical models can never be obviated and undervalued.

Table 1. Sentence Variations in English and Bengali

English	Bengali
How are you?	Apni kemon achen? (Formal Style)
	Tumi kemon acho? (Semi-Formal Style)
	Tui kemon achish? (Informal Style)

Researchers have taken keen interest on building personalized language models for different purposes. In 2009, Xue et al. proposed a method for personalizing search results based on user interest [1]. They modeled individual profile using statistical language models, and finally constructed clusters to form group models. The models incorporated in a cluster were gathered from people who have same taste in web content. Li et al. used statistical language models for personalizing information extraction services *i.e.* text snippet extraction [2].

The existing methodologies for personalized word prediction emphasizes mostly on estimation based upon the built-in statistical language models, which are consisted of using the dictionary of a particular language. However, for phonetic typing the task is not that simple, as the spelling of a word may differ from user to user and deviate from the standard form, if one is considered as standard. Apart from this, several other issues compound the task, and we look forward to develop a personalized next word predictor as a remedy to this situation. Based on the problems mentioned above, we would like to address the following research questions in this work :

- Will personalized language models improve the next word prediction component used by an IM?
- How the sources of data for training language models effect personalization?
- How can we measure the performance improvement of such systems by creating a proper data set and evaluation strategy?

In order to answer these questions, we would like to develop person specific statistical language models, of which one will be invoked, when a person is communicating with another person. Till this end, we have come up with two hypotheses:

- A single user follows a specific linguistic style while communicating with another person
- Excluding data that degrades a language model, can improve the performance of the model in the context of improving the next-word prediction component of an IM service

Based on these two assumption we modeled specific persons style by building statistical language models with an exclusion method. Thus, the key contributions of this work are :

- Building language models following a users linguistic style especially in Bengali. The linguistic style is captured based on the interaction of a user with other users.
- An exclusion method based language model which would exclude the unnecessary information from the model and would produce better suggestions for user.

2 Related Works

In this section, we review the background literature related to our personalized next word prediction strategy. A generalized word prediction system was proposed by Bosch, which could predict millions of words per second [3]. He used a simple decision-tree algorithm that was less costly in terms of complexity, in order to use a large amount of data for training from the *Reuters* corpus. However, a personalization component was not included in this method, as it was not developed considering the dynamics of human communication. Siska et al. designed an adaptive keyboard that could adjust its predictive features and key displays based on current user input [4]. They implemented the personalized word prediction module using common English dictionary to improve the performance of such a system. The built-in English dictionary was used with an existing database that the system needed to overwrite personalized phonetic words. Nonetheless, this method requires a huge database of training corpora which is not suitable for a smart-phone based implementation.

A learning approach employing hierarchical modeling of phrases was proposed by Richard et al. [5]. This approach reduced the amount of initial training data required to facilitate on-line personalization of the text prediction system.

It is also intended for the development of assistive technologies for disabilities, especially within the domain of augmentative and alternative communications (AAC) devices. The key insight of the proposed approach is the separation of stop words, which primarily play syntactical roles in phrases. Matthew suggested a system to improve the rate at which users can participate in a conversation using an AAC (Augmentative and Alternative Communication) device. This was intended for persons who are unable to communicate verbally [6].

Author profiling techniques were also used for personalizing messaging systems, and most of these systems are based on machine learning approaches. Tayfun et al. proposed to investigate the possibility of predicting several users and message attributes in text-based, real-time, on-line messaging services [7]. Specifically, they aimed to identify instant message authors correctly using style-based approach. Inches et al. designed a framework for identifying topic and author from on-line user-generated conversations [8]. They used different similarity metrics to identify document features and took an entropy-based approach to identify authors. Author identification have been improvised a step further by Villatoro-Tello et al. where they identified misbehaving authors in instant messaging by classifying user text and building models based on SVM and neural networks [9].

Sarwar et al. showed that constructing a LM with the conversation text pair of users, and trying to predict the text of other users provides different outcomes for different users. Even though it seems quite intuitive, the outcome of this research indicated that a LM built on a conversation text could be useful to predict the text of a cluster of users [10].

3 Background

In this section we explain two necessary topics that are essential to our proposed method: *language model* and *perplexity*.

3.1 Language Model

Language models (LM) are heavily used in many applications using Machine Translation and Speech Recognition technology. Language models are used to evaluate the probability of a sequence of words. Given a sequence of words of length m, it is possible to estimate the probability of the sequence $P(w_1, w_2, ..., w_m)$, using LM [11]. Based on the context there are different types of LM. If the probability of a word w_k, depends on its previous word w_{k-1}, then it is denoted as bi-gram LM. However, in general LM are defined as n-gram language model, where the probability $P(w_1, \ldots, w_m)$ of observing the sentence $w_1, \ldots, w_m$ is approximated as shown in Equation 1.

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \tag{1}$$

The joint probability distribution can be estimated as below:

$$\prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) = \frac{count(w_1, \ldots, w_{i-1}, w_i)}{count(w_1, \ldots, w_{i-1})} \tag{2}$$

In case of bigram language model Equation 2, can be re-written as Equation , based on markov assumption.

$$\prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) = P(w_1) \prod_{i=2}^{m} P(w_i \mid w_{i-1}) \tag{3}$$

In these paper we have used bigram language model to extract the linguistic style of an author.

3.2 Perplexity

Perplexity is a measure that is used to test the quality of LM. In order to test LM, test data is used and perplexity is measured. Let us assume that there are m sentences in test data: $t_1, t_2, \ldots, t_m$. It is possible to measure the log probability of each sentences using LM:

$$\log \prod_{i=1}^{m} P(t_i) = \sum_{i=1}^{m} logP(t_i) \tag{4}$$

Now, Perplexity (PP) can be defined using the following equation:

$$PP = 2^{-l}, \; where \quad l = \frac{1}{M} \sum_{i=1}^{m} logP(t_i) \tag{5}$$

in 5, M is the total number of words in the test data. The lower the value of perplexity the better the LM are. The worst possible LM results in the number of words in the test data. Perplexity is a measure of effective *branching factor* [12].

4 Proposed Method

The proposed method is developed based on the hypothesis that *A single user follows a specific linguistic style while communicating with another person.* Thus, at first, our aim is to construct a collection of personalized dictionaries *i.e.* language models for a user. Finally, those models would be used to predict the next word for the user at the time of sending instant messages. The invocation of a specific model would be completely dependent on the person, with whom the user will be communicating.

Language models can assign a probability value to a word given a sequence of words. For example, using *bigram* language model, we can predict the probability of a word "computer" given the word "personal". Moreover, using language model notation, we can represent it as $P(computer \mid personal)$. Using

this value, we want to estimate the probability of the word "computer", given the word "personal". To describe our method, we use the terms *language models* and *models*, interchangeably. In the following paragraphs, we would discuss our methodology from the perspective of a single user (u), for whom we would build a set of models (M), which will facilitate his computer-mediated communication with other users (U) in his network. Moreover, there would be a one-to-one relationship between M and U, *i.e.* $|M| = |U|$.
In order to describe the method, we consider that user u has connection with a set of k users $U = \{u_1, u_2, u_3, \ldots, u_k\}$, through an instant messaging service. Interaction set $I = \{i(u, u_1), i(u, u_2), \ldots, i(u, u_k)\}$ contains all the messages sent to each $u_k \in U$ by user u. Hence, we are only considering the unidirectional messages sent by user u to all other users. According to our own definition these messages form a General Dictionary (GD_u), which we use to build a generalized model for u.

According to the first part of our research hypothesis, GD_u can not be a suitable source of observed data to build a generative model, which can be used to predict the chat content of u and u_k. As the instant messaging content of a user varies significantly, based on the other person he is communicating with, GD_u would be a source of data that would degrade the model. Some conversations in GD_u can lead to the development of inefficient models, and building a next word predictor based on those models would not improve the communication speed. Thus, in order to model the interaction $i(u, u_k)$, we would need a distinct model $m(u, u_k)$, and it should be built on $I_k \subset I$. This would result in a model set $M = \{m(u, u_1), m(u, u_2), \ldots, m(u, u_k)\}$. Now, when u would be communicating with u_k, $m(u, u_k)$ would be invoked to generate words for u.

The second part of our hypothesis is about the construction of I_k. If we exclude a subset of interactions $\bar{I}$ from I, we would be able to get textual contents that model the conversation between u and u_k more closely. Thus, we can construct I_k using the following equation:

$$I_k = I - \bar{I} \tag{6}$$

From equation 6, it can be seen that $\bar{I}$ is a cluster of interactions, which we will exclude from I. Our goal is to construct $m(u, u_k)$ using I_k. In order to build $m(u, u_k)$, we construct one model for each interaction from $I - i(u, u_k)$. After that we evaluate the perplexity of each model on the held out data from interaction $i(u, u_k)$. After that we select top-n models that result in highest perplexity values, and create interaction set $\bar{I}$, by including the associated interactions with them. We also refer $\bar{I}$ as the Worst Interaction Set (WIS) for the ease of understanding. Thus, we are trying to estimate, which models maximize the uncertainty, while predicting the held out data. We hypothesize that the associated interactions used to build these models introduce more uncertainty in GD_u. By excluding $\bar{I}$ from I, and constructing a model on I_k, we reduce the entropy in GD_u. As a consequence, the final model $m(u, u_k)$ would be a better predictor than a generalized model constructed from GD_u.

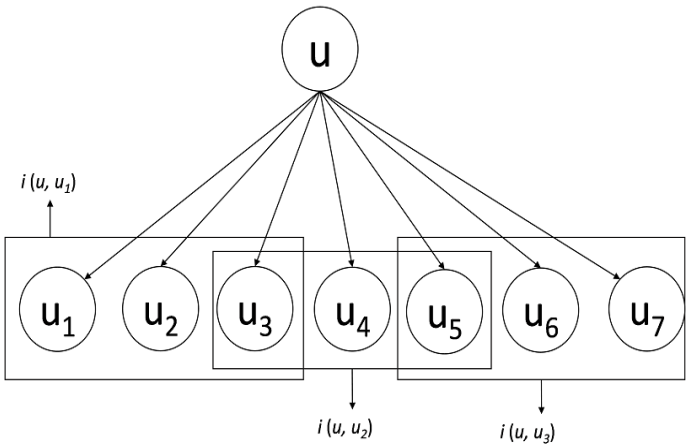

Fig. 1. Interaction clustering based on top-3 interactions

We have shown a sample execution of our proposed method using Figure 1. Initially, we create a LM based on $i(u, u_1)$ and evaluate the perplexity of the model using the Equation 5, for all the users $u_j \in U$. As a result, for each $u_j \in U$, we get a perplexity value. The worst perplexity of a LM on a test data is the number of words in the test data. According to the scope of our work, we only consider bigram based LMs.

After obtaining the perplexity values for each $i(u, u_j) \in I$, we sort them in descending order and select the top-n interactions. We extract the user id $u_t \in U$ from the interactions and create a user group with those values. We perform this process repeatedly by building LM with all the interactions from I one by one. From Figure 1, it can be observed that for interaction $i(u, u_4)$, three interactions have been grouped together: $i(u, u_1)$, $i(u, u_2)$ and $i(u, u_3)$. As these three interactions produced three highest values of perplexity with the LM constructed using $i(u, u_4)$, they are grouped together.

5 Experimental Setup

5.1 Data Set

We have collected the summary and analysis using our program from the chat logs of three different facebook users. Chatting data is completely private and we did not collect the data from users, instead we provided our program to the users and they gave us the output generated from the program. All the users were IT professionals; they could run our program to generate summary data for us. Each of the users, who ran our program, communicated with at least 7 different people and their basic interaction was in Bengali; the total collection

contained 22 interactions. Prior to running our program, a privacy agreement was signed by each user.

Testing and training data set was created from the chat logs by our script. Last 20% data of each interaction of a user was kept as test data. We trained our model on the first 80% data and evaluated the model on the last 20% held out data. In table 2, some properties of our data set are shown formally.

Table 2. User chat log data properties.

User	Average sentence length per line	Total words per interaction	No of Interactions
101	14	5466	7
102	18	3555	8
103	24	5782	7

It can be seen from Table 2 that we have given each user a unique identifier, so that he or she can be remained anonymous. According to the table, User 101 has 14 sentences on an average in each interaction set, with a total of 5466 words. It is also evident that user 101 interacted with 7 persons in total. Each individual user was asked to provide his messaging content considering different groups of people like family, friends, cousins, colleagues etc. so that we can get different types of interactions.

5.2 Experimental Setup and Result

In this work, we have tried to select the best model that performs well in terms of perplexity, on the held out data of a specific user interaction. At first, we create a generalized bigram model over all the user interactions I. In this paper, we use the term General Dictionary (GD) in exchange with I for the ease of understanding. After the general bigram model is created, we use it to evaluate the performance of GD for all the interactions of a specific user. Then we create a specialized LM, namely WIM for each interaction by subtracting $\bar{I}$ from I. For the experimentations in this paper, we subtract top-3 interactions from GD, and build models on resultant data. Finally, each of the models is evaluated based on the calculation of perplexity on the held out data of each user interaction. The percentage improvement of WIM with respect to GD is calculated and shown in all our result tables. A negative value depicts poor performance of WIM, whereas a positive value represents performance improvement. The complied results from the experimentation for each user are shown using Table 3, Table 4 and Table 5, respectively.

From Table 3, we can see that user 101 interacts with a total of 7 persons with 7 different ID's. For interaction (101,201) its worst interaction set WIS consists of the user with ID's 202, 203, 206. This means that these interactions actually

degrade the performance of the language model built for user 101, using the GD. Here, the WIM improves the model by 9.49% which is considerably higher than GD. However, in interaction (101,206) we can see that WIM actually gives 18.6% poor result comparing to GD. It is observed that those cases are very rare, when GD outperforms WIM.

Table 3. Different LM result on user 101 chat log.

Interaction (u_i,u_j)	WIM	WIM Perplexity	GD Perplexity	Improvement of WIM over GD(%)
(101,201)	{(101,202),(101,203),(101,206)}	14.27	15.76	9.49
(101,202)	{(101,205),(101,207),(101,206)}	16.89	19.280	12.36
(101,203)	{(101,202),(101,205),(101,206)}	18.81	21.36	11.95
(101,204)	{(101,201),(101,202),(101,206)}	17.83	19.39	8.00
(101,205)	{(101,202),(101,207),(101,206)}	19.72	21.47	8.17
(101,206)	{(101,203),(101,205),(101,204)}	31.77	26.78	-18.66
(101,207)	{(101,205),(101,202),(101,206)}	15.21	17.22	11.66

Table 4. Different LM result on user 102 chat log.

Interaction (u_i,u_j)	WIM	WIM Perplexity	GD Perplexity	Improvement of WIM over GD(%)
(102,301)	{(102,308),(102,306),(102,304)}	30.04	12.35	-143.28
(102,302)	{(102,308),(102,306),(102,304)}	21.37	23.79	10.18
(102,303)	{(102,306),(102,307),(102,30)8}	11.47	12.82	10.51
(102,304)	{(102,305),(102,308),(102,306)}	12.46	13.59	8.36
(102,305)	{(102,301),(102,308),(102,304)}	10.49	12.01	12.68
(102,306)	{(102,305),(102,301),(102,304)}	12.61	14.32	11.90
(102,307)	{(102,301),(102,306),(102,304)}	12.19	14.70	17.10
(102,308)	{(102,305),(102,304),(102,307)}	11.74	13.85	15.26

In table 4, the interaction between user 102 and other users are shown. Here, we can see that while interacting with user 301, WIM gives worse result comparing to GD. In all the other cases, WIM performs significantly better than GD.

Table 5 shows the performance on the interactions of user 103. In the interactions (103,402), (103,403), (103,404) WIM performs poorly giving the improvement percentage -13.57%, -79.64%,-13.63% respectively. However, in the interaction (103,403) the result is very poor in comparison with GD.

Table 5. Different LM result on user 103 chat log.

Interaction (u_i,u_j)	WIM	WIM Perplexity	GD Perplexity	Improvement of WIM over GD(%)
(103,401)	{(103,406),(103,404),(103,402)}	10.16	11.83	14.10
(103,402)	{(103,405),(103,404),(103,403)}	12.34	10.87	-13.57
(103,403)	{(103,406),(103,407),(103,405)}	18.11	10.08	-79.64
(103,404)	{(103,405),(103,402),(103,406)}	13.31	11.71	-13.63
(103,405)	{(103,406),(103,401),(103,404)}	10.55	11.87	11.14
(103,406)	{(103,403),(103,404),(103,405)}	8.98	9.86	8.93
(103,407)	{(103,404),(103,402),(103,405)}	8.49	10.32	17.73

In our experiment, we have shown that the language models built by excluding the Worst Interaction Set (WIS) from I improves the performance of the general dictionary based LM. By excluding WIS, we actually remove the contents, which affect the performance. However in some cases, we have found that excluding WIS from I doesn't always improve the performance; in fact in some situations GD outperforms WIM.This phenomenon occurs, because we have subtracted a fixed number of interactions from GD for our experiment. Moreover, there are some interactions in WIS cluster, which might generate important suggestions for user. By excluding them, we are removing those important information from GD, which results in poor perplexity scores. As a result, it can be experimentally inferred that excluding WIS from the interaction set will build better LM than the LM built over the generalize dictionary for a single user. But, in this paper, we have conducted small experimentation, and publish the results after running our program with input from three users only. Therefore, even though the results are quite interesting, we can not finally conclude that excluding information from the GD of a user will model her conversation more accurately.

6 Conclusion

The research leads to the development of a user-oriented and personalized next-word predictor for instant messaging, which can speed up the text-based communication among different people in the virtual world. The ever-growing field of social media and instant messaging have created the necessity to design a system that could support fast, comfortable and smooth typing. Even though we have shown our result in terms of a standard NLP metric, perplexity, we hope to implement an instant messaging system for the on-line evaluation of our idea. Moreover, we would like to collect more user chat log with privacy agreement, anonymize our data set using some well known anonymization algorithms like k-anonymization and publish our data set in future. Besides, we would try to filter out some unnecessary information *i.e* emoticon, stop words, punctuation marks etc. which will improve the performance of the language models.

Acknowledgment This work is supported by the University Grant Commission, Bangladesh under the Dhaka University Teachers Research Grant No-Regi/Admn-3/2016/46897.

References

1. Gui-Rong Xue, Jie Han, Yong Yu, and Qiang Yang. User language model for collaborative personalized search. *ACM Transactions on Information Systems (TOIS)*, 27(2):11, 2009.
2. Qing Li and Yuanzhu Peter Chen. Personalized text snippet extraction using statistical language models. *Pattern Recognition*, 43(1):378–386, 2010.
3. Antal Van Den Bosch. Scalable classification-based word prediction and confusible correction. *Traitement Automatique des Langues*, 46(2):39–63, 2006.
4. Siska Fitrianie and Leon Rothkrantz. An adaptive keyboard with personalized language-based features. In Vaclav maousek and Pavel Mautner, editors, *Text and Speech and Dialogue 2007*, number LNAI. Springer, Springer, sep 2007.
5. Richard Gabriel Freedman, Jingyi Guo, William H. Turkett Jr., and Victor Pal Pauca. Hierarchical modeling to facilitate personalized word prediction for dialogue. In *AAAI Workshop: Plan, Activity, and Intent Recognition*, volume WS-13-13 of *AAAI Workshops*. AAAI, 2013.
6. Matthew E. J. Wood. Syntactic pre-processing in single-word prediction for disabled people. Technical report, Bristol, UK, UK, 1996.
7. Tayfun Kucukyilmaz, B. Barla Cambazoglu, Cevdet Aykanat, and Fazli Can. Chat mining: Predicting user and message attributes in computer-mediated communication. *Inf. Process. Manage.*, 44(4):1448–1466, July 2008.
8. Giacomo Inches and Fabio Crestani. Online conversation mining for author characterization and topic identification. In *Proceedings of the 4th workshop on Workshop for Ph. D. students in information & knowledge management*, pages 19–26. ACM, 2011.
9. Esaú Villatoro-Tello, Antonio Juárez-González, Hugo Jair Escalante, Manuel Montes-y Gómez, and Luis Villasenor Pineda. A two-step approach for effective detection of misbehaving users in chats. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.
10. Sheikh Muhammad Sarwar and Abdullah-Al-Mamun. Next word prediction for phonetic typing by grouping language models. In *2016 2nd International Conference on Information Management (ICIM)*, pages 73–76. IEEE, 2016.
11. Jianfeng Gao, Joshua T. Goodman, and Jiangbo Miao. The use of clustering techniques for asian language modeling, 2001.
12. Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity - a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.