I.J. Intelligent Systems and Applications, **2015, 01, 94-100**
Published Online December 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2015.01.09

A Stochastic Prediction Interface for Urdu

Qaiser Abbas
Fachbereich Sprachwissenschaft, Universität Konstanz, Konstanz, 78457, Germany
Email: qaiser.abbas@uni-konstanz.de

Abstract—This work lays down a foundation for text prediction of an inflected and under-resourced language Urdu. The interface developed is not limited to a T9 (Text on 9 keys) application used in embedded devices, which can only predict a word after typing initial characters. It is capable of predicting a word like T9 and also a sequence of word after a word in a continuous manner for fast document typing. It is based on N-gram language model. This stochastic interface deals with three N-gram levels from unary to ternary independently. The uni-gram mode is being in use for applications like T9, while the bi-gram and tri-gram modes are being in use for sentence prediction. The measures include a percentage of keystrokes saved, keystrokes until completion and a percentage of time saved during the typing. Two different corpora are merged to build a sufficient amount of data. The test data is divided into a test and a held out data equally for an experimental purpose. This whole exercise enables the QASKU system outperforms the FastType with almost 15% more saved keystrokes.

Index Terms—Urdu Prediction Interface, N-Gram Language Model, QASKU, Word and Sequence Prediction, Corpus Based Application

I. Introduction

In this modern era, the management of time becomes an important skill and people want to finish their work as early as possible with quality and quantity. This theory applies on every walk of our life and the same is true for writing, synthesizing and identification in a document [22, 26] of a word processing application or searching a query in a web browser. Today, the word processing applications, text editors, web browsers and spam detection [21], etc., on our machines are satisfying human typing needs quite efficiently and the facility of character/word/sentence prediction and recognition [23] becomes a tool to reduce the time of human typing on machines. There is a large prediction support existed on machines for English language and also for the other European languages as well. These prediction tools include stand-alone tools like AutoComplete by Microsoft, AutoFill by Google Chrome, TypingAid[1] and LetMeType[2]. Free wares include short hand tools, context completion tools, line completion tools[3], etc. As Urdu is an under-resourced language and no such precise or sufficient support available on today's machines for this language. This work presented here is a positive contribution for Urdu word prediction (UWP) in general and a helpful tool to boost up the typing needs of related handicapped persons.

A number of techniques are in use for word prediction in Natural Language Processing (NLP). The Prediction Suffix Tree (PST) model is one of them, which was claimed that this was the best ever strategy existed. In which, an efficient data structure along with a Bayesian approach was introduced. It was used to maintain the tree mixtures. These mixtures had a better performance than any other model, provided that the weights for the mixtures were efficiently selected [11, 12] along with the Bayesian framework [19]. This mixture theory was used on different corpora and was observed that it was much better theoretically and practically than the N-gram model [16]. To boost up insertion, deletion and search process in this PST model, the splay trees [18] were used. The Bayesian model helped to evaluate two priorities. First, it defined a probability distribution recursively over all the PSTs and secondly, it observed the probability of the word appeared for the first time in a given text. It included two possibilities further, a simply new word or a word previously observed but not in this context. This problem was solved through the use of Good Turing algorithm [20]. This model had only one failure, which was in the context of syntactic and semantic information. The purpose of presenting an introduction of this PST model is given next.

The adaptation of PST Model during QASKU's (a name proposed for author's work) construction was an ideal state but at that time, the PST model could not be applied due to non-availability of the resources like Urdu Treebank. In near future, the extension of this QASKU model towards the PST's approach will be possible because a treebank for the Urdu language is under construction by Abbas [1]. So, it was decided to move forward in consecutive steps. As a first step, the N-gram approach had been adopted in the construction of this model. Moreover, the reason for the selection of N-gram approach was this that the N-gram could be converted into PSTs but vice versa was not possible as concluded in [16]. A description of the N-gram approach used in the QASKU model is given in Section II-A. This N-gram based language model lays down the foundation for the QASKU's future work and also for the UWP. The QASKU process is divided into four sub-processes labeled as 1 to 4 in Fig 2. The details are presented in Section II-B.

The evaluation of the QASKU was done using the measures introduced by Aliprandi [5]. These measures or metrics include the Keystroke Saved effort percentage

[1] http://www.autohotkey.com/community/viewtopic.php?f=2\&t=53630
[2] http://www.clasohm.com/lmt/en/
[3] http://jsimlo.sk/notepad/

(KS), Keystrokes Until Completion (KUC) and the percentage of Time Saved during Word typing (WTS). The respective measuring formulas are discussed in the beginning of section III. Basically, these measures were applied on the FastType model for Italian. This was also a N-gram based model along with some extra features. Both Urdu and Italian are different languages in their structure and orthographic nature, but the N-gram's approach adopted is independent of language nature. A performance comparison of QASKU and the FastType was made and presented in section IV. FastType model was first introduced by Aliprandi [4] as an algorithm of linear combination for Italian word prediction. This algorithm was first enhanced with a subsystem called DonKey for the human interaction interface [6]. In recent years, a further extended model of the FastType came up with statistical and rule based approach for the word prediction. FastType was built to predict words mainly for the inflected and also for non-inflected languages [5]. FastType used mainly the N-gram model for word prediction and its Keystroke Saving (KS) is 51%. The model used the parts of speech (POS) [13] and the morpho-syntactic information for presenting a list of words [5]. This enrichment of linguistic information is necessary for getting precise and accurate prediction results for the inflected languages.

Two different corpora were merged and used for the evaluation of the QASKU model. One of which is known as the Urdu 5000 most frequent words [14], which is available at the website [4] of the Center for Language Engineering, Pakistan and the other is 10M words raw Urdu corpus developed by Raza [17]. Further discussion on the corpora is presented in section II.A and finally, the future work and conclusion are presented in Sections V and VI respectively.

II. Design

A. N-gram acquisition

As the word prediction applications for the English language are concerned, a number of approaches and algorithms have been existed and exploited. However, the word prediction in Urdu has not yet been fully utilized or exercised in any general or commercial level product except some applications like T9 on embedded devices for which the word prediction for the Urdu language is existed just for the sake of presence and nothing more. Mostly, the successful word prediction systems used N-gram model to predict words. So, in this QASKU model, a same approach in contrast of models discussed in Section I has been adopted due to non-availability of resources. As we had to evaluate uni, bi and tri grams probabilities first, So, the uni-gram probability values were calculated computationally from the available merged corpus using the formula given in (1) as follows, where *Wi* is an individual word and *TW* is the total numbers of words in a corpus.

$$P(W_i) = \left(\frac{Count(W_i)}{Count(T_w)}\right) \quad (1)$$

The term merged corpus means a merger of the 5000 most frequent words of Urdu collected from a 19.4 million words corpus and the 1 million (M) words corpus extracted from a 10M raw Urdu newspaper data. This merger was used specially in the case of uni-gram probability calculation simply to increase the size of the final corpus. Unfortunately, the document of the Urdu 5000 most frequently used words contained only counts of words. So, the probability value for each word listed in the document was calculated using (1). A sample of uni-gram and probability values is given in Fig. 1, which is a sample picture for the database of the QASKU interface labeled as level 1 in Fig. 2. The dashes at the end of the Fig. 1 means, the unique words with probable values continued in the respective columns of the database. In contrast of the uni-gram, only 1M portion of the corpus was used to evaluate the bi-grams and the tri-grams because the document of 5000 most frequent words contained only the unique uni-gram counts and no any raw text of Urdu. Each uni, bi and tri gram data of the corpus was divided into training and the test data individually according to the standard division of 80% and 20% respectively. Training data and the 10% of test data (held out data) were recorded in the QASKU's database for training purpose. The held out data was kept along with the training data, just to measure the difference between the results of held out data and the test data. This difference was expected due to small size (1M + 5000 words) of the merged corpus.

Similarly, the bi-gram probability values as shown in Fig. 1 were calculated computationally using a formula given in the (2), where *Wi* and *Wi-1* are the next and previous words respectively.

$$P\left(\frac{W_i}{W_{i-1}}\right) = \left(\frac{Count(W_{i-1}W_i)}{Count(W_{i-1})}\right) \quad (2)$$

Finally, the tri-gram probability values shown in Fig. 1 were also evaluated using the same formula as mentioned in (2) except an addition of one more word *Wi-2* from the history. Equation (3) depicts the calculation of the tri-gram probabilities.

$$P\left(\frac{W_i}{(W_{i-1}W_{i-2})}\right) = \left(\frac{Count(W_{i-2}W_{i-1}W_i)}{Count(W_{i-2})}\right) \quad (3)$$

The basic understanding for obtaining the probabilities of uni, bi and tri grams are described above and further detail & history of the N-gram language model adopted can be seen in [15].

B. QASKU Process Model

An interface of the QASKU model was developed and the model had three approaches of prediction. First is the uni-gram mode, second is the bi-gram mode and third is the tri-gram mode. Architecture and flow of the QASKU process model is given in Fig. 2. The whole process is divided into four sub processes (levels) labeled as 1 to 4. A sub process labeled 1 is a process in which a user has

[4] http://www.cle.org.pk

to select an option from the given levels of prediction e.g., uni-gram, bi-gram or the tri-gram. After the selection, the relevant database of the N-gram is loaded in the memory of the machine with the most probable N-gram. In 2 and 3 as depicted, when a user types a letter from the keyboard, then this typed letter goes directly into the *Main* process. It is then matched with the loaded probable N-gram selected. The uni-gram probable list of words can update and reduce itself automatically in the list box with respect to letters/words typed in the text box. In case of bi-gram or tri-gram level of prediction, a single word or double words are typed respectively into the text box and then the list box is updated and reduced with predicted words automatically with respect to the typed letter of the word. When the space key is pressed by the user, the predicted word displayed in the list box is replaced with the incomplete/complete typed word in the text box and the complete predicted word/words are stored/updated into another main textbox used for the collection of words/sentences. All N-gram prediction levels have an iterative mechanism of processing which is achieved by updating the predicted word from the list box to textbox and then by storing the predicted word/words into the main textbox. Due to this iterative mechanism, the QASKU is really helpful in writing long articles/ documents. The switching between these three levels of N-gram is explicit and can be activated at any time by simply clicking on the relevant option e.g. uni, bi or tri grams.

Uni-Gram	Probability	Bi-Gram	Probability	Tri-Gram	Probability
پولیس	0.002198647	پولیس کے	0.130044843	پولیس کے ہاتھوں	0.137931034
کے	0.041961627	کے ہاتھوں	0.002819549	کے ہاتھوں ظلم	0.083333333
ہاتھوں	0.000187329	ہاتھوں ظلم	0.052631579	ہاتھوں ظلم و	1
ظلم	0.000216907	ظلم و	0.545454545	ظلم و زیادتی	0.166666667
و	0.00215921	و زیادتی	0.00913242	و زیادتی کی	0.5
زیادتی	0.000187329	زیادتی کی	0.157894737	زیادتی کی خبروں	0.333333333
کی	0.029381027	کی خبروں	0.00033557	کی خبروں نے	1
خبروں	9.8594E-06	خبروں نے	1	خبروں نے لوگوں	1
نے	0.010815767	نے لوگوں	0.001823154	نے لوگوں کا	1
لوگوں	0.00098594	لوگوں کا	0.1	لوگوں کا اعتماد	0.2
کا	0.015489125	کا اعتماد	0.001273074	کا اعتماد مجروح	0.5
اعتماد	9.8594E-05	اعتماد مجروح	0.1	اعتماد مجروح کیا	1
مجروح	1.97188E-05	مجروح کیا	0.5	مجروح کیا ہے	1
کیا	0.004900124	کیا ہے	0.120724346	کیا ہے ۔	0.633333333
ہے	0.023761166	ہے ۔	0.592946058	ہے ۔ پولیس	0.007697691
۔	0.044702542	۔ پولیس	0.007498897	۔ پولیس کو	0.058823529
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....

Fig. 1. Probability values for uni, bi and tri grams

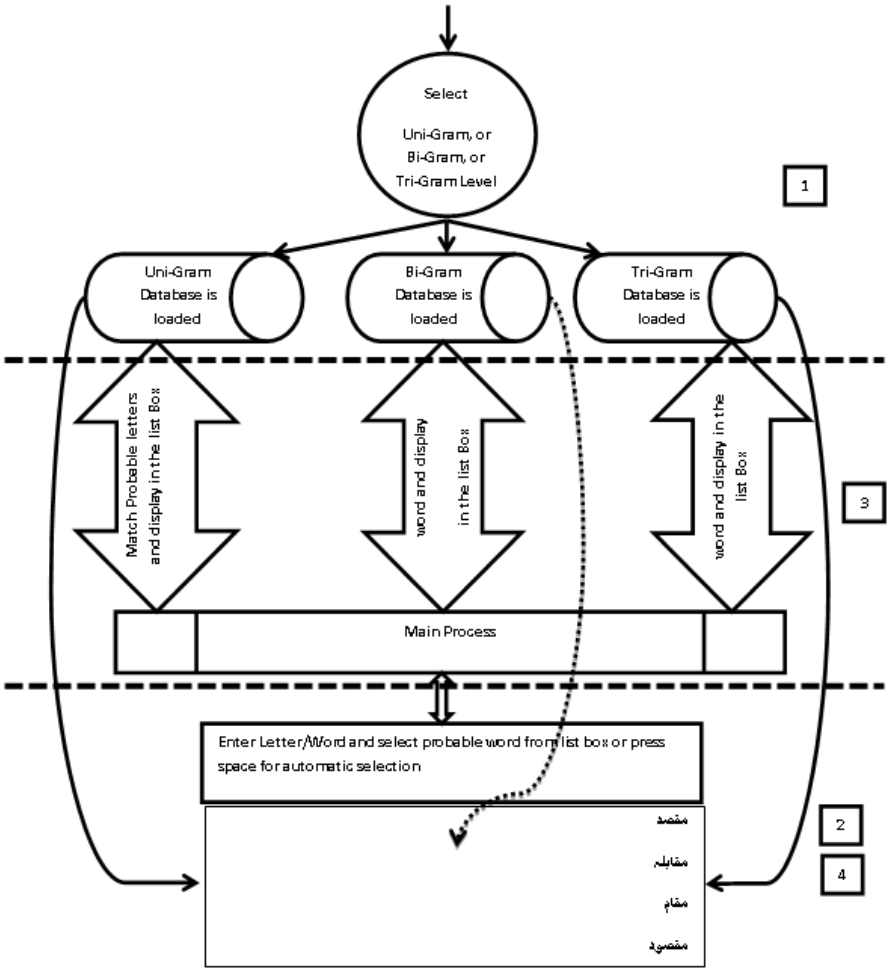

Fig. 2. The architecture and flow of the QASKU process model

III. Evaluation and Results

The *KS* is calculated with the following formula mentioned in [5], in which *KT* is the total number of keystrokes required to type the text and *KE* is the effective number of keystrokes during typing of the text.

$$KS = \left(\frac{K_T - K_E}{K_T}\right) \times 100 \quad (4)$$

Other measures used to calculate the performance are the *KUC* and the *WTS*, which were used by Aliprandi [5].

$$KUC = \left(\frac{(C_1 + C_2 + C_3 + \cdots + C_n)}{n}\right) \quad (5)$$

Here $C_1 + C_2 + C_3 + \cdots C_n$ are the number of keystrokes required to type each of the n words until the correct version of the word appears in the list box. While in the *WTS*, which is the percentage of the time saved during the typing, *Tn* is the total time required in typing the text without using the QASKU model and *Ta* is the time consumed in typing the same text by using the QASKU model. The formula is given in (6).

$$WTS = \left(\frac{T_n + T_a}{T_n}\right) \times 100 \quad (6)$$

Two different performance evaluation trials had been performed on uni, bi and tri grams with respect to parameter L, which is the length of the predicted text in characters including the typed keystrokes. These trials were performed on different lengths ranging from 15 to 50. Performance results of the test data and the held out data are given in Table 1 and Table 2 respectively. The tables contained independent results of the *KS*, *KUC* and the *WTS* for uni, bi and tri grams respectively. The discussion and issues of these trials is presented in Section IV.

In order to judge the overall performance of the QASKU model including all uni, bi and tri grams, the following sample text in Fig. 3 was typed and predicted. The translation of the text from Urdu to English is irrelevant here and hence avoided. The keystrokes/characters highlighted gray in the text is the typed data, while the un-highlighted text is the predicted data by the QASKU model. The total number of characters or keystrokes (*KT*) in this text is 288 and the effective number of keystrokes (*KE*) is 100. By using (4), the overall *KS* of the QASKU model achieved is 65.28%. All three uni, bi and tri grams modes have an equal explicit share in this *KS* calculation, otherwise the highest bi-gram *KS* percentages of this model are 70.23% and 72.47% for the test and the held out data respectively, which can be seen in Table 1 & 2.

Table 1. Performance evaluation with 10% test data

	Uni-Gram	Bi-Grams			Tri-Grams		
	L<= 15	L<=20	L<=30	L<=40	L<=30	L<=40	L<=50
KS	52.77%	34.61%	66.66%	70.23%	49.46%	55.17%	60.25%
KUC	5.02	7.84	10.01	12.46	15.86	18.15	20.13
WTS	32.08%	26.23%	44.73%	57.50%	33.75%	41.64%	46.77%

Table 2. Performance evaluation with 10% held-out data

	Uni-Gram	Bi-Grams			Tri-Grams		
	L<= 15	L<=20	L<=30	L<=40	L<=30	L<=40	L<=50
KS	54.14%	35.56%	69.59%	72.47%	51.42%	58.72%	65.05%
KUC	4.96	7.25	9.24	11.07	12.63	14.43	15.94
WTS	33.80%	28.83%	48.37%	59.10%	37.08%	45.12%	52.36%

IV. Discussion and Issues

As a work in the domain of the inflected Urdu language, the QASKU is a little behind than the other prediction models discussed in Section I. QASKU handles uni gram to meet the requirements of the standalone applications like T9, AutoComplete, AutoFill, etc., and similarly handles bi & tri grams to meet the requirements of the applications like sentence/context completion, line completion, etc., only for the fast typing. Evaluated results of the QASKU are compared with the results of the FastType [5] model and presented as follows.

At first, the performance is evaluated on the test data. At uni-gram level of prediction, the average *KS* obtained is 52.77% which is almost equal to 53.14% of the FastType when L<=15. At bi-grams, the average *KS* obtained is 34.61%, the *KUC* is 7.84 and the *WTS* is 26.23% for L<=20, which is 16% less in the *KS*, 5.82 number of keystrokes ahead in the *KUC* and 2.92% extra time consumption during typing than the FastType. This concludes that the FastType has better results as compared to the QASKU. However, when the length is increased to L<=40, the QASKU gives the average *KS* equals to 70.23%, the *KUC* equals to 12.46 and the *WTS* equals to 57.50% which is 20% better in the *KS*, 27.50% more time saving *WTS* value and the *KUC* consumes 10.46 more keystrokes, which concludes a great advantage of the QASKU over the FastType. The detail of results can be seen in Table 1. Similarly, at tri-gram level of prediction, the QASKU predicts better when the desired text becomes lengthy. Moreover, the FastType results were evaluated only for L<=20 while the QASKU is evaluated up to L<=50 maximum. Even it is able to work for L>50.

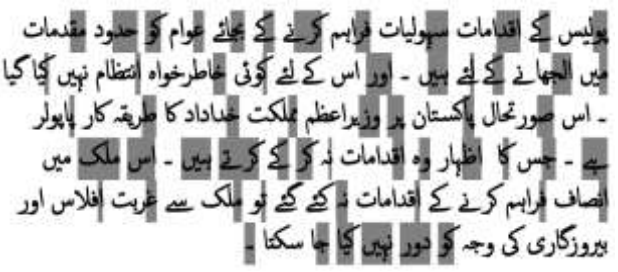

Fig. 3. Prediction results of the QASKU model

At second, the performance is evaluated with the 10% held out data. The results are shown in Table 2. The QASKU is trained on the training and the held out data sets conditionally as discussed in Section II.A. In this case, with the uni-gram level of prediction, the QASKU beats the FastType with almost 1% in the average *KS* and

almost 5% more time saving in the *WTS* but again lagging behind the FastType with almost 3 more keystrokes consumed in the case of the *KUC*. At bi & trigrams levels of prediction, again the QASKU outperforms the FastType for a lengthy text prediction with almost 2% to 5% more average *KS*, respectively as compared to the test data. Similarly, almost 2% to 6% increase has been achieved in the *WTS*. The *KUC* is unbelievably too high as compared to the FastType with a continuous reduction in evaluation.

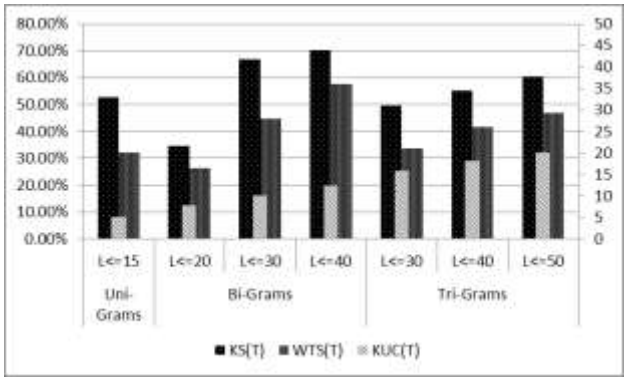

Fig. 4. Performance evaluation on 10% test data

As the same measures used with the FastType were taken for the performance evaluation. The QASKU predicts almost equal in case of uni, bi and tri grams when the length L is kept less than or equal to 20. However, when the length of text is raised beyond 20, then the QASKU started its influence over the FastType's results very rapidly and the whole story becomes in the favor of the QASKU model. The respective comparison charts of the test and the held out data are depicted in Fig. 4 and Fig. 5 respectively.

In both the figures, the percentages are given on the left side y-axis while the right side y-axis contained the *KUC* values. All three levels of the predictions (uni, bi and tri grams) are given on the x-axis along with their respective maximum length size L, for which the performance is evaluated. A comparison of *KS*, *WTS* and *KUC* on the test and the held out data has also been performed and represented in Fig. 6, Fig. 7 and Fig. 8 respectively.

There is no big difference in the case of uni-gram. However in the case of bi-gram and particularly in the trigram, when the length L of the predicted text was increased, the percentage of the *KS* was also increased from 2% to 2.5% for the held out data which further concluded that the QASKU performed better with the increase in length L. In Fig. 7, the KS (T) and the KS (H) are the keystrokes saved for the test and the held out data, respectively. This difference in percentage between the test and the held out data can be reduced by increasing the size of the corpus because greater the size of the corpus reduces the probability of the unknown words in the test dataset. In case of the uni-gram, a very small difference in the *KS* might be due to the result of the corpora merger between the 5000 most frequent words and 1M raw Urdu words as discussed in Section II.A. Figure 7 for the WTS has a continuous increase from the unigram to the tri-gram level and the ratio of this increase is around 3% to 6%, which concluded that the QASKU prediction is helpful in reducing the time during the typing of the text.

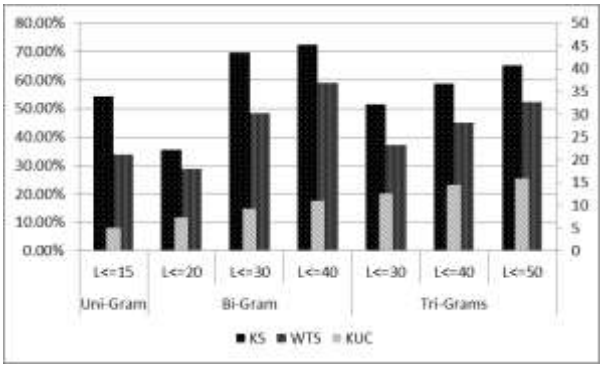

Fig. 5. Performance evaluation on 10% held out data

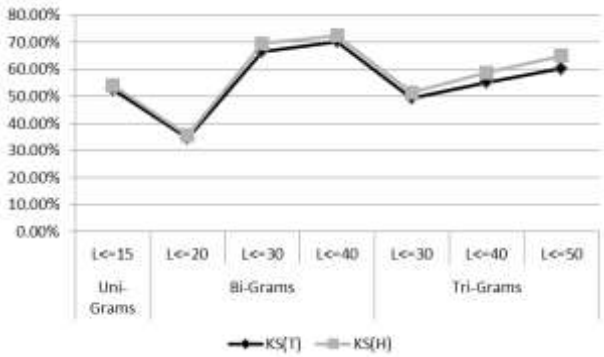

Fig. 6. A comparison of KS between the test and held out data

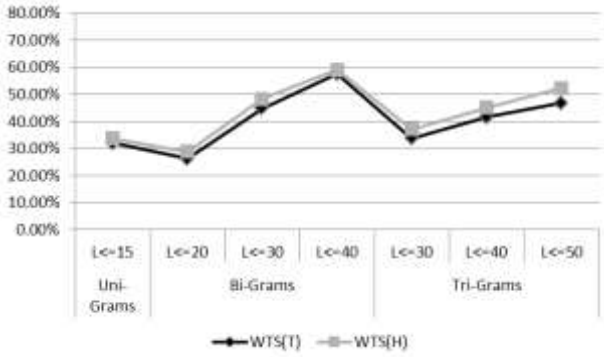

Fig. 7. WTS comparison between test and held out data

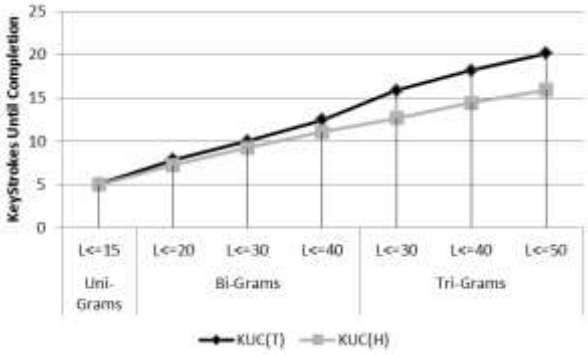

Fig. 8. KUC comparison between test and held out data

Figure 8 represents the comparison of the *KUC* between the test and held-out data. It shows that the *KUC* of the held-out data is lower than the *KUC* of the test data. It means that in case of the held-out data, a less number of keystrokes are consumed until completion of the desired result. The distance between the two lines is also giving us a conclusion that the uni-gram approach for standalone applications like T9 is appropriate due to less number of keystrokes for an inflected Urdu language. As this approach cannot be adopted in other applications like sentence/context completion, line completion, etc. So, after performance evaluation of the bi and tri grams, it is concluded that the bi-gram approach with less number of keystrokes as compared to the tri-gram is more beneficial for such type of applications. This cannot be claimed fully until an experiment on a different corpus cannot be performed.

Despite of all these experimental results, an overall performance is evaluated as mentioned in Section III. Figure 3 represents the rows of a sample text executed on the QASKU. The output contained 288 numbers of total keystrokes *KT*, in which 100 is the effective numbers of keystrokes *KE* and 188 is the numbers of predicted keystrokes. Thus, the overall *KS* percentage achieved is 65.28%, which is almost 15% more than the overall *KS* percentage of the FastType model for Italian.

V. Future Work

At the end, as mentioned in Section I that the linguistic information like the Urdu POS [25], morphological information [8, 9, 10], Urdu Ezafe [7], verb morphological forms [2, 24], etc., can improve the prediction results. The enrichment of linguistic information is possible in the extended PST version of the QASKU model because the resources needed are in pipeline. On the other hand, the embedding of such useful information from an Urdu treebank mentioned in Section I along with the NU-FAST treebank [3] is still in progress and publically not available. This information will really be useful in providing a boost to the QASKU. A robust searching algorithm is also the main task for the future extensions of this work. The two important issues discussed in section IV are in need of some serious effort for its solution. One is the emergence of the high value of *KUC* during the performance evaluation of the held out data and the second is to conclude a suitable option between bi and tri grams for the prediction of the Urdu sentences.

VI . Conclusion

This model helps the handicapped people to type fast just like normal human being and also strengthens the normal ones further ahead. The QASKU with overall 65.28% of *KS* is comparable or better than the state of the art resources in the domain of Urdu language and is a positive contribution in Urdu language processing. This model has a quality of being more efficient with the increase in length L of the text, which is quite good in case of inflected languages like Urdu. Its performance can be enhanced after encoding linguistic information

References

[1] Q. Abbas, "Building a Hierarchical Annotated Corpus of Urdu: The URDU.KON-TB Treebank", Lecture Notes in Computer Science (LNCS), Vol. 7181(1), P 66-79, ISSN 0302-9743, Springer-Verlag Berlin/Heidelberg, 2012.

[2] Q. Abbas and A. N. Khan, "Lexical functional grammar for Urdu modal verbs" In Proceedings of 5th IEEE International Conference on Engineering and Technology (ICET), 2009.

[3] Q. Abbas, N. Karamat and S. Niazi, "Development Of Tree-Bank Based Probabilistic Grammar For Urdu Language" International Journal of Electrical & Computer Science, Vol. 9(09), pp. 231–235, 2009.

[4] C. Aliprandi, N. Carmignani, and P. Mancarella, "An Inflected-Sensitive Letter And Word Prediction System", International Journal of Computing & Information Sciences, Vol. 5(2), pp. 79-85, 2007.

[5] C. Aliprandi, N. Carmignani, N. Deha, P. Mancarella, and M. Rubino, "Advances In NLP Applied To Word Prediction", Langtech, 2008.

[6] C. Aliprandi, N. Carmignani, P. Mancarella, and M. Rubino, "A Word Predictor For Inflected Languages: System Design And User-Centric Interface", In Proceedings of the 2nd IASTED International Conference on Human-Computer Interaction, March 2007.

[7] T. Bögel, M. Butt, and S. Sulger, "Urdu Ezafe And The Morphology-Syntax Interface", In proceedings of LFG08, 2008.

[8] M. Butt, and T. Ahmed, "The Redevelopment Of Indo-Aryan Case Systems From A Lexical Semantic Perspective", Morphology, Vol. 21(3-4), pp. 545-572, 2011.

[9] M. Butt, and T. H. King, "Restriction For Morphological Valency Alternations: The Urdu Causative" Intelligent linguistic architectures: Variations on themes by Ronald M. Kaplan, pp. 235-258, 2006.

[10] M. Butt, and G. Ramchand, "Complex Aspectual Structure In Hindi/Urdu" M. Liakata, B. Jensen, & D. Maillat, Eds, pp. 1-30, 2001.

[11] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How To Use Expert Advice", Journal of the ACM (JACM), Vol. 44(3), pp. 427-485, 1997.

[12] A. DeSantis, G. Markowsky, and M. N. Wegman, "Learning Probabilistic Prediction Functions", In 29th Annual Symposium on IEEE, Foundations of Computer Science, pp. 110-119, October 1988.

[13] A. Fazly, and G. Hirst, "Testing The Efficacy Of Part-Of-Speech Information In Word Completion", In Proceedings of the 2003 EACL Workshop on Language Modeling for Text Entry Methods, Association for Computational Linguistics, pp. 9-16, April 2003.

[14] M. Ijaz, "Urdu 5000 Most Frequent Words", Technical report, Center for Research and Urdu Language Processing, National University of Computer & Emerging Sciences, Lahore, PK, 2007.

[15] C. D. Manning, and H. Schütze, Foundations Of Statistical Natural Language Processing, Cambridge: MIT press, 1999.

[16] F. C. Pereira, Y. Singer, and N. Tishby, “Beyond Word N-Grams”, In Natural Language Processing Using Very Large Corpora, pp. 121-136, Springer Netherlands, 1999.
[17] G. Raza, Sub-categorization Acquisition And Classes Of Predication In Urdu, PhD Thesis, 2011.
[18] D. D. Sleator, and R. E. Tarjan, “Self-Adjusting Binary Search Trees”, Journal of the ACM (JACM), Vol. 32(3), pp. 652-686, 1985.
[19] F. M. Willems, “The Context-Tree Weighting Method: Extensions”, IEEE Transactions on Information Theory, Vol. 44(2), pp. 792-798, 1998.
[20] I. J. Good, “The Population Frequencies Of Species And The Estimation Of Population Parameters”, Biometrika, Vol. 40(3-4), pp. 237-264, 1953.
[21] Jaber Karimpour,Ali A. Noroozi,Adeleh Abadi, "The Impact of Feature Selection on Web Spam Detection", IJISA, vol.4, no.9, pp.61-67, 2012.
[22] Souleymane KOUSSOUBE, Roger NOUSSI, Balira O. KONFE, "Using Description Logics to specify a Document Synthesis System", IJISA, vol.5, no.3, pp.13-22, 2013.DOI: 10.5815/ijisa.2013.03.02
[23] Leandro Luiz de Almeida, Maria Stela V. de Paiva, Francisco Assis da Silva, Almir Olivette Artero, "Super-resolution Image Created from a Sequence of Images with Application of Character Recognition", IJISA, vol.6, no.1, pp.11-19, 2014. DOI: 10.5815/ijisa.2014.01.02
[24] Q. Abbas, G. Raza, “A computational classification of Urdu dynamic copula verb”, International Journal of Computer Applications (IJCA), Vol. 85 (10), pp. 1-12, ISSN: 0975 - 8887, Published by Foundation of Computer Science, New York, USA, 2014.
[25] Q. Abbas, “Semi-Semantic Part of Speech Annotation and Evaluation”, In Proceedings of ACL 8th Linguistic Annotation Workshop held in conjunction with COLING, Association of Computational Linguistics, P 75-81, Ireland, 2014.
[26] Q. Abbas, M. S. Ahmed, S. Niazi, “Language Identifier for Languages of Pakistan Including Arabic and Persian”, International Journal of Computational Linguistics (IJCL), Vol. 01(03), P 27-35, ISSN 2180-1266, 2010.

Authors’ Profiles

Qaiser Abbas. Did master in Computer Science from NUCES (National University of Computer and Emerging Science), Lahore, Pakistan. Going to complete my doctorate in Computational Linguistics or Natural Language Processing from University of Konstanz, Germany in August 2014. Working as a research assistant in University of Konstanz, Germany. Working as a lecturer in University of Sargodha, Pakistan at Department of Computer Science, since 2003. The detailed profile along with the research work can be seen on the following URL http://ling.uni-konstanz.de/pages/home/abbas

How to cite this paper: Qaiser Abbas,"A Stochastic Prediction Interface for Urdu", International Journal of Intelligent Systems and Applications (IJISA), vol.7, no.1, pp.94-100, 2015. DOI: 10.5815/ijisa.2015.01.09

