Australian Journal of Basic and Applied Sciences, 5(5): 1137-1143, 2011
ISSN 1991-8178

Probabilistic Analysis of Sindhi Word Prediction using N-Grams

Javed Ahmed Mahar, Ghulam Qadir Memon

Faculty of Engineering, Science and Technology, Hamdard University, Karachi, Pakistan

Abstract: Word prediction is an essential subtask of various Natural Language Processing (NLP) applications of Arabic scripting based languages like Persian, Urdu, and Sindhi; especially it is necessary for diacritic restoration systems. Sindhi is highly homographic language and text is written without diacritic symbols in everyday life applications therefore, word prediction is a difficult task. The task of word prediction is investigated by using three N-gram models i.e., bigram, trigram and 4-gram that approximates the probability in given history of words, these models are trained on corpora of Sindhi language. Statistical NLP is always based on corpora because word frequencies are calculated from this corpus therefore for training and testing the corpus containing approximately 3 million token are used. Add-one smoothing technique is used to assign non-zero probabilities to all N-grams having zero probabilities. The performance of N-gram models are measured by using Entropy and Perplexity models, the achieved results show clearly that the 4-gram language model is more suitable for Sindhi language because it has lower perplexity value than other N-grams. Some of languages as Arabic, Persian and Urdu have the same characteristics as in Sindhi for the reason probabilistic analysis may be useful for NLP applications of above mentioned languages on same scale.

Key words: Word Prediction; Language Model; N-Gram; Perplexity; Corpora.

INTRODUCTION

Sindhi is highly homographic language like in Arabic and text is written without diacritics in real life applications like newspapers, magazines, books etc. that creates lexical and morphological ambiguity, for instance, a word سر can be written by using diacritic symbols as /Sir⍰/ سرَ (a brick) (noun), /Sir⍰/ سرُ (the head) (noun), /S⍰r⍰/ سُرُ (tone) (noun), /S⍰r⍰/ سُرَ (tones) (noun), /S⍰rr/ سُرّ (move) (verb/noun), /S⍰r⍰/ سَرُ (a kind of reed) (noun), /S⍰r⍰/ سَرَ (a string of beads) (noun), /S⍰ri/ سَر (head, principal) (noun), /Sirr/ سرّ (secret) (noun), /Siri/ سرِ (at, on, upon) (adverb), it shows that one word have several word types and each word represents different meaning (Mahar and Memon, 2009). Due to the absence of diacritic symbols, word prediction is a difficult task and decreases predictability in language modeling therefore looking at previous word can give us sufficient indication about what the next ones are going to be.

Statistical language modeling is ever a challenging task for morphologically rich languages (Duh and Kirchhoff, 2004); such languages have large number of word types in a given text. The N-grams are probabilistic models widely used for word prediction and provide some directions to assign probabilities to words; in this study the purpose of language modeling is to predict the next word (w_n) in a sentence using the previous

w_1^{n-1} words. N-gram represents an Nth order Markov language model (Jurasfky and Martin, 2000) derived from

large number of training corpus and rely on the likelihood of word sequences, for instance, word pairs (bigram), word triples (trigram) and word fourth (4-gram), these models are used for the probabilistic relationship between words. The bigram, trigram and 4-gram models have been implemented and evaluated extensively on Sindhi corpora, by calling this L. The Add-One smoothing technique is used to assign non-zero probabilities to all N-grams having zero probabilities for increasing the performance of N-grams. Perplexity is a way of evaluating language modeling in NLP (Brants *et al*., 2007) and it is computing on the average size of the word set over correctly recognized words, perplexity is defined as 2 raised to power of entropy ($2^{H(corpus}$),

Corresponding Author: Javed Ahmed Mahar, Faculty of Engineering, Science and Technology, Hamdard University, Karachi, Pakistan.
E-mails: mahar.javed@gmail.com, gqmemon@yahoo.com; Ph: +92-334-2727937;
Fax: +92-243-9280439

the perplexity of models is used for comparing same vocabulary, in this connection, a model having low value is considered the best one.

Over the last two decades, statistical techniques have been used in various NLP applications like speech recognition and machine translation (Kim and Khudanpur, 2003) and were also applied for language modeling. N-grams are useful for computational linguistics and applied mathematics, (Paskin, 2001) presented a probabilistic model of syntax which is based on grammatical bigrams and were trained on raw text using an EM algorithm. (Siivola and pellom, 2005) showed how an n-gram model can be built by adding suitable sets of n-grams to a unigram model until desired complexity is reached; they compared their growing method to entropy based pruning. An algorithm presented by (Zhu *et al*, 2008) that learns bigram language models from bag-of-words and used EM algorithm that seeks a model which maximizes the regularized marginal likelihood of the bag-of-words documents.

Some languages have rich morphological structure like in Arabic, Urdu, French, Persian, Sindhi etc, the language modeling is mandatory for these languages for the task of word prediction, the comparative study of Arabic and French is investigated by (Meftouh and Smaili, 2009) uses n-gram models, the performance is measured with different smoothing techniques, they reported that the trigrams models are appropriate for French with any smoothing technique and higher order smoothed n-gram model with Witten Bell method are efficient for Arabic. Text prediction techniques are commonly used to enhance the communication rate in augmentative and alternative communication, a survey on text prediction techniques is presented by (Vitoria and Abascal, 2006), they examined prediction applications and related features like block size, dictionary structure, prediction method and user interface, prediction measurement parameters and published results are also compared.

2. N-Gram in Word Prediction:

N-gram based language models estimate the probability of occurrence for a word, given a string of *n*-1 preceding words. The process of computing probability of words helps us for prediction of word in a sequence and we can find out the next word in a sequence (Jurafsky and Martin, 2000). The input text is a sequence of

words can represent as: <$w_1, w_2, w_3, ...w_n$ *or* w_1^n >, if each word is occurring in its appropriate position then probability can represent as:

$$P(w_1,\ w_2,\ w_3, ...,w_{n-1},\ w_n) \tag{1}$$

After using chain rule of probability for decomposition:

$$P(\ w_1^n\ ) = P(w_1)P(\ w_2|\ w_1)\ P(\mathrm{w}_3|\ w_1^2\ )\ \ldots\ldots\ P(\mathrm{w}_\mathrm{n}|\ w_1^{n-1}\ )$$

$$= \prod_{k=1}^{n} P(w_k \mid w_1^{k-1}) \tag{2}$$

Computing the probability of a word given a long sequence of preceding words like $P(\mathrm{w}_\mathrm{n}|\ w_1^{n-1}\ )$ is a

difficult task therefore, we first approximate the probability of a word given all the previous words using bigram model by the conditional probability of the preceding word $P(\ w_n \mid w_{n-1})$. The bigram model is a first-order Markov model because it looks one token into the past can be generalized to the trigram (second-order) because it looks two token into the past and thus to N-gram (N-1th order) because it looks N-1 tokens into the past), the general equation is:

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1}\ ) \tag{3}$$

For a bigram grammar, we compute the probability of a complete string by substituting eq. (3) into eq. (2). The result:

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k \mid w_{k-1}) \tag{4}$$

The trigram and 4-gram are same as bigram except the condition on two and three previous words respectively. As bigram matrix is too spare and having large number of zero probabilities so that we need to assign a non-zero probability to zero probability bigrams. One smoothing ways is to add one to bigram counts before normalizing; this method gives us a useful baseline (Jurafsky and Martin, 2000), many smoothing estimates rely on an adjusted count C^* the adjusted count for add-one smoothing is defined as:

$$C_i^* = (C_i + 1)\frac{N}{N + V} \tag{5}$$

If we will not received any evidence from N-1th order N-gram then we use back-off to a lower order –gram, for example, for 4-gram if $C(w_{i-3}w_{i-2}w_{i-1}w_i) > 0$ then we use $P(w_i \mid w_{i-3}w_{i-2}w_{i-1})$ (4-gram) else if $C(w_{i-3}w_{i-2}w_{i-1}w_i) = 0$ and $C(w_{i-2}w_{i-1}w_i) > 0$ then we use $P(w_i \mid w_{i-2}w_{i-1})$ (trigram) else if $C(w_{i-2}w_{i-1}w_i) = 0$and $C(w_{i-1}w_i) > 0$then we use $P(w_i \mid w_{i-1})$(bigram) otherwise we use $P(w_i)$ (unigram).

3. *Methodology:*

The methodology for word prediction of Sindhi text is mainly consists on five phases. (1) For training and testing, the corpus is collected from different genres like books, newspapers and magazines through Internet, downloaded files are converted into plain text files, spelling mistakes and the absence of short vowels from text is corrected manually. (2) Generally, words are delimited through white spaces like in English but many words are present in Sindhi lexicon having inter-space in the words, therefore, we have implemented tokenization algorithm for segmentation of Sindhi words (Mahar and Memon, 2010). (3) Three types of N-grams: Bigram, trigram and 4-gram models are implemented by using eq.3 for computing the probability of words, in this connection, three databases are designed and each N-gram results are stored into separate database. (4) Add-one smoothing technique is used (see eq.5) for assigning non-zero probabilities to all N-grams having zero probabilities for getting better results. (5) Finally, the accuracy of N-gram models in terms of word prediction of Sindhi text is compared with perplexity measurement.

3.1. Corpus Description:

The corpus of any language is important for statistical natural language processing applications especially for getting probabilities of words in a sequence. The proposed word prediction system is relies on statistical methods that has been trained on a large corpus of data. For training and testing, five types (arts, sports, politics, environment, music) of corpus L were collected from different genres like newspapers, magazines and books from Internet, the L contained approximately 3 million tokens, the pages were downloaded in HTML and PDF formats, after downloading pages are converted into their plain text equivalents. The L is divided into two parts: (i) $L^{training}$ that contains 2924967 word tokens and 338831 word types. We carefully designed our $L^{training}$ and tried to make it generalized, (ii) L^{test} that contains 135362 word tokens and 26426 word types which is approximately 4.62% of training data. The detailed statistical information of training and testing data is shown in Table 1. All inflected words are treated as separate words because N-grams are based on word forms, the words are counted for computing their probabilities.

Table 1: Statistical information of proposed training and testing corpus.

Set	Corpus Type	Sentences	Word Token	Word Types
Training	Arts	31157	685504	48416
	Sports	22184	421496	38317
	Politics	40333	847071	65164
	Environment	33900	640104	49546
	Music	18544	330792	37088
Testing	Arts	1300	27003	3901
	Sports	1250	25882	2482
	Politics	1350	28350	3543
	Environment	1400	26600	3325
	Music	1250	27527	3175

3.2. Tokenization:

The tokenization is the first process of our proposed mechanism; it is the process of segmenting input sequence of orthographic symbols, the division of input text into tokens is necessary for language modeling. As the orthography of Sindhi is based on the concatenation of syllables and words are normally delimited through white spaces like in English, but segmentation of words sometimes may be ambiguous due to presents of inter-hard space in a single word, for instance, a word /Sahibe q⍰dir⍰ti/ صاحب قدرت is a compound word and we explicitly put hard space between / Sahibe/ صاحب and / q⍰dir⍰ti/ قدرت, this explicit hard space is affecting on tokenization process, therefore, a new tokenization scheme for Sindhi text was implemented (Mahar and Memon, 2010).

4. Experiments and Results:

For experiments, we have performed some preprocessing operations on proposed corpus because during the files conversion process we observed that many words have spelling mistakes and necessary short vowel symbols are absent from text. As the orthographical structure of Sindhi language is highly variable therefore these mistakes were corrected manually.

Sindhi script is written from right to left; the cursor is flashed from right to left direction. For Sindhi writing system, we install MB Sindhi software (Majid, 2006), for correct representation and visualization of words MB Lateefi is selected as default font style. The tokenization algorithm (Mahar and Memon, 2010) is implemented for correct segmentation of sequence of words. Many words are present in our corpora having high relative frequencies of occurrence. For example, a word /maan/ مان /I/ occurs 863 times and a word /Sin⍰⍰/ سنڌ (Sindh) occur 1351 times, by contrast a word /raandi/ راند (game) occurs only 4 times, these frequencies have been used to assign a probability distribution across following words. The N-gram probabilities are come from the training corpus, for prediction of next word in a Sindhi sentence, the previous words are used to assign probabilities to words, for instance, consider the sentence

سنڌ جي تاريخ پنج هزار سال پراڻي آهي
/Sin⍰⍰ ⍰e tarex p⍰n⍰⍰ h⍰zar⍰ saal⍰ p⍰ra⍰i aahe/
[is][old][years][thousand][five][history][of][Sindh]
(The history of Sindh is five years old.)

has the following word level trigrams.

سنڌ جي تاريخ	/Sin⍰⍰ ⍰e tarex/	[history] [of] [Sindh]
جي تاريخ پنج	/⍰e tarex p⍰n⍰⍰/	[five] [history] [of]
تاريخ پنج هزار	/tarex p⍰n⍰⍰ h⍰zar⍰/	[thousant] [five] [history]
پنج هزار سال	/p⍰n⍰⍰ h⍰zar⍰ saal⍰/	[years] [thousant] [five]
هزار سال پراڻي	/h⍰zar⍰ saal⍰ p⍰ra⍰i/	[old] [years] [thousant]
سال پراڻي آهي	/saal⍰ p⍰ra⍰i aahe/	[is] [old] [years]

Bigram assign higher probabilities to those pairs having sensible relation like /a⍰san ⍰⍰/ اسان جو (ours), /Si⍰⍰ l⍰⍰⍰e/ سج لٿي (at a sun set), /⍰⍰n⍰⍰ girh⍰⍰i/ چنڊ گرهڻ (moon eclipse), Sin⍰⍰ ⍰⍰⍰rite/ سنڌ ڌرتي (Sindh's earth) , /k⍰mput⍰r Saainsi/ ڪمپيوٽر سائنس (computer science) and assign lower probabilities to those pairs having non-sensible relation like /v⍰U⍰ ⍰⍰⍰kir⍰/ ويو ڇوڪرو (boy gone out), /a⍰⍰⍰e veh⍰/ اڙي ويهه (eh! Sit), /⍰⍰⍰k⍰ a⍰⍰⍰⍰ie/ ٿڪ اٿئي (spit on you), /a⍰⍰⍰ h⍰⍰⍰⍰⍰/ اک هڻڻ (wink of eye), /bahi SOri/ باه سور (kindle fire), the sample of words bigrams are shown in Table 2.

Table 2: The sample of Sindhi words bigrams.

Words	Bigrams	Words	Bigrams
بيشڪ الله	21	بيٽ بال	7
وڏيڪ عزت	16	وڻ پوکڻ	14
سڀ ڪجھ	39	جي لاءِ	46
اهم اصول	13	تقرير لاءِ	6
مٺي مانٽي	12	جنهن لاءِ	43
سندن لاءِ	26	۾آهن	53
قديم دور	17	انهيءَ لاءِ	58
اهڙي ريت	11	گندي سياست	10
آسماني ڪتاب	17	سانوگاه	15
سنڌ ڌرتي	36	ماني ڪاءُ	11

The experiments are performed on our developed Sindhi corpora, for testing we have randomly selected 15,000 sentences, 3000 sentence from each corpus type and then applied N-gram grammars. The backoff smoothing algorithm is used for estimating the N-gram Probabilities. By using training and testing corpus we have evaluated three types of N-gram models which are bigram, trigram and 4-gram. Each N-gram results are stored into a separate database. The calculated results of each N-gram on Sindhi corpus are shown in Table 3.

Table 3: Calculated results of Sindhi N-grams

Corpus Type	Bigram	Trigram	4-gram
Arts	8323	10441	12945
Sports	11917	18099	22721
Politics	13554	20661	25988
Environment	6998	12927	13778
Music	12338	17207	20166
Total	53130	79335	95598

The N-gram models are trained by counting and normalizing and then computed the probability of randomly selected sentences. Two types of sentences were used for testing. (a) the test sentences those are already included in our training corpus , therefore, these sentences have high non-zero probabilities (b) the test sentence those are not included in our corpus, therefore, these sentences have zero or low non-zero probabilities. The average size of sentence is 14 words and the average word length is 5 letters. As the training corpus is sparse and large number of cases has zero probability especially in bigrams. Therefore, we have reevaluated this and assign non-zero probabilities to all bigrams having zero probabilities by using Add-One smoothing technique, these models have been trained on sentences having different context.

4.1. Performance Measures:

An N-gram model is compared in terms of accuracy to find out which model is more suitable for test set of Sindhi language by using perplexity. The results which we have received during our experiments on test corpus shows that the 4-gram models are considered best as compared to others, this selection is based on perplexity measurements. The perplexity measurement of our test corpus according to N-gram grammars is shown in Table 4 and the graphical representation of perplexity measurement of three N-gram models is depicted in Figure 1. We have computed perplexity for comparing probabilistic models on our test set of 135362 words; the perplexity is defined as 2^H.

Table 4: Performance of N-grams in terms of perplexity.

N-gram Order	Entropy	Perplexity
Bigram	6.78	109.89
Trigram	6.28	77.71
4-gram	6.11	69.07

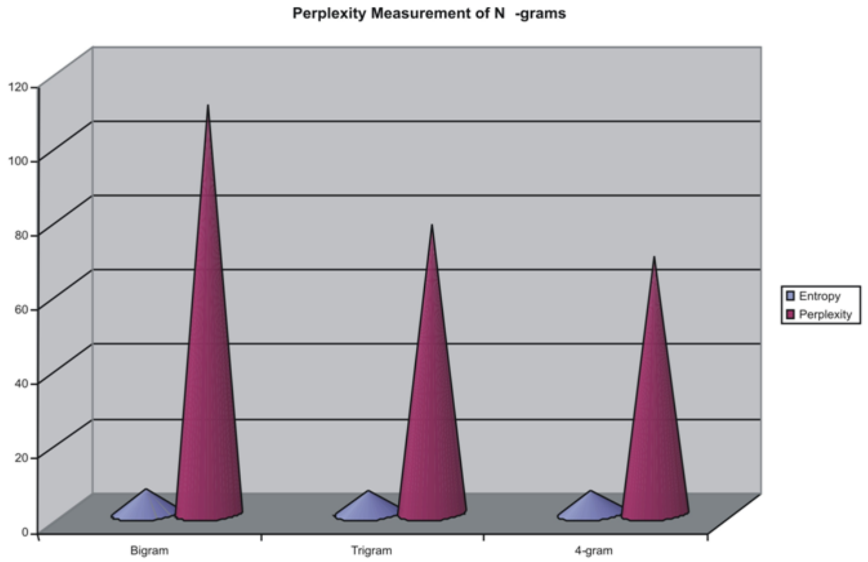
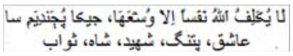


Fig. 1: Perplexity measurements of three N-grams

Discussions:

Word prediction is important for various computational linguistic applications like part of speech tagging, word sense disambiguation, speech recognition, spelling error detection, augmentative communication and diacritic restorations. Particularly, the efficiency of word prediction system is measured for the task of diacritic restorations of Sindhi text. The purpose of this analysis is to develop a word prediction system which is based on statistical language modeling, the performance of only three N-Gram models i.e., bigram, trigram and 4-gram is investigated for word prediction of Sindhi language and according to perplexity values it is observed that 4-gram is sufficient for Sindhi word prediction system, it may be possible that better results can be achieved with 5-gram or 6-gram. We select Sindhi language because many words of Arabic, Urdu and Persian are present in Sindhi, for example, consider one line from our proposed corpus from right to left six words are used from Quran (a holy book), many words like are same in Urdu and Sindhi, similarly few words Sindhi are common with Persian, we have also achieved acceptable results of these languages therefore our proposed analysis for word prediction may be used for Arabic, Urdu and Persian.



لا يُكلِفُ اللهُ نفساً إلا وُسْعَهَا، جيڪا پُڇنديم سا
عاشق، پتنگ، شهيد، شاه، ثواب

Various smoothing algorithms like Add-One, Witten-Bell Discounting, Good-Turing Discounting, Katz etc were used to assign non-zero probabilities to all N-grams having zero probabilities; Add-one smoothing technique is used because it is simple to implement, the other smoothing algorithms will used for analyzing the variation of different N-gram results. As Sindhi is a syllable based language, therefore, morpheme based language modeling is recommended instead of whole word. This study intends to use N-grams for Syllable based Sindhi language modeling and this analytical data of N-grams will used for probabilistic Sindhi POS tagging and diacritic restoration systems.

Conclusions:

It is successfully implemented and evaluated a word prediction system based on statistical language modeling for the Sindhi language. In this paper, only three N-gram language models: bigram, trigram and 4-gram are investigated for the task of word prediction of Sindhi language. The Add-one smoothing technique is used to assign non-zero probabilities to all N-grams having zero probabilities. The words are segmented by using our developed tokenization algorithm. Orthographically, Sindhi language is very similar with Arabic, Urdu and Persian languages; therefore this probabilistic analysis may be used for these languages. The results showed that the 4-gram model is more efficient for Sindhi language compare to lower level models. The corpora of 3 million words are developed for experiments. The corpus of Sindhi language is used in the context of statistical approaches to Sindhi diacritic restoration project.

REFERENCES

Brants, T., A.C. Popat, P. Xu, F.J. Och, J. Dean, 2007. “Large Language Models in Machine Translation”, in: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, pp: 858-867.

Duh, K., K. Kirchhoff, 2004. “Automatic Learning of Language Model Structure”, in: Proceedings of the 20th International Conference on Computational Linguistic, Geneva, pp: 148-154.

Jurafsky, D., J.H. Martin, 2000. “Speech and Language Processing: An Introduction to Natural Language Processing”, Computational Linguistic and Speech Recognition, *Prentice-Hall*.

Kim, W., S. Khudanpur, 2003. “Cross-Lingual Lexical Triggers in Statistical Language Modeling”, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Morristown, USA, pp: 17-24.

Mahar, J.A., G.Q. Memon, 2009. “Phonology for Sindhi Letter to Sound Conversion”, Journal of Information and Communication Technology, 3(1): 11-20.

Mahar, J.A., G.Q. Memon, 2010. “Rule Based Part of Speech Tagging System for Sindhi Language”, in: Proceedings of the International Conference on Signal Acquisition and Processing, IEEE Computer Society Press, Bangalor, India, pp: 101-106.

Majid, B., 2006. MB Sindhi software downloaded from www.fileguru.com/apps/mb-sindhi-software.

Meftouh, K., K. Smaili, 2009. “Comparative Study of Arabic and French Statistical Language Models”, in: Proceedings of the International Conference on Agents and Artificial Intelligent, Porto, Portugal, pp: 156-160.

Paskin, M.A., 2001. “Grammatical Bigrams”, in: T. Dietterich, S. Becker, and Z. Gharahman, editors, Advances in Neural Information Processing Systems (NIPS) 14, MIT Press.

Siivola, V., B.L. Pellom, 2005. “Growing an N-gram Language Model”, INTERSPEECH, Lisbon, Portugal, pp: 1309-1312.

Vitoria, N.G., J. Abascal, 2005. “Text prediction systems: a survey”, Universal Access in the Information Society, 4(3): 188-203.

Zhu, X., A.B. Goldberg, M. Rabbat, R. Nowak, 2008. “Learning Bigrams from Unigrams”, in: the Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technology,Columbus, OH.