N-gram Based Word Sense Disambiguation of Hindi Post Position से (sē) in the context of Hindi to Punjabi Machine Translation System

Rakesh Kumar
Research Scholar, Punjab Technical University, Kapurthala (India) E-Mail: rakesh77kumar@yahoo.com

Vishal Goyal
Dept. of Computer Science, Punjabi University, Patiala (India), E-Mail: vishal.pup@gmail.com

Ravinder Khanna
Dean (R&D), MM University,Sadopur, Ambala (India), E-Mail: ravikh_2006@yahoo.com

Abstract
India has many regional languages. Attempts have been made for developing machine translations between these languages, but little success has been reported so far. Analysis of Hindi to Punjabi machine translation system devised by Punjabi University, Patiala, India has found that Hindi post position से (sē) is translated inaccurately being its ambiguous nature, most of the times, as it has eighteen different senses in Punjabi. The overall translation success rate of this system reported as 87.60%, however the translation success rate in respect of this post position से (sē) is only about 2%.

In this paper, N-gram approach (along with its smoothing variants) has been applied to improve the accuracy of translation of this post position से (sē) in already developed Hindi to Punjabi Machine Translation System. It has been concluded that bigram approach with Add-One smoothing algorithm gives the best results in improving the accuracy of translation of post position से (sē) from 2% to 85.49%, thus improving the overall machine translation accuracy of the system from 87.60% to 92.30% .

Keywords*: Natural Language Processing (NLP), Word Sense Disambiguation (WSD), Machine Translation (MT)*

1. Introduction

Word Sense Disambiguation (WSD) is one of the most challenging areas of Natural Language Processing (NLP). Many words have multiple meanings in almost all natural languages for example in the following English language sentence:

(a) That bird is **crane**.

(b) They had to use a **crane** to lift the object.

In the first sentence the word **'crane'** is a bird while in the second it is a lifting machine. WSD methods are extensively used in Machine Translation (MT) to disambiguate words having multiple meanings in the target language. Hindi (the national language) to Punjabi (a language widely spoken in North West India and eastern Pakistan) machine translation system of Punjabi University, Patiala, India [1] claims a success rate of about 87.60%. However the Hindi post position से (sē) when translated using this system poses a major difficulty. This post position has eighteen different senses in Punjabi (see Figure 1). Selecting the correct sense of this post position in Punjabi has been a source of error and has lowered the success rate of the system. In this paper, various N-gram WSD methods and their smoothing variants have been applied to improve the viability of translation of the system [2,3].

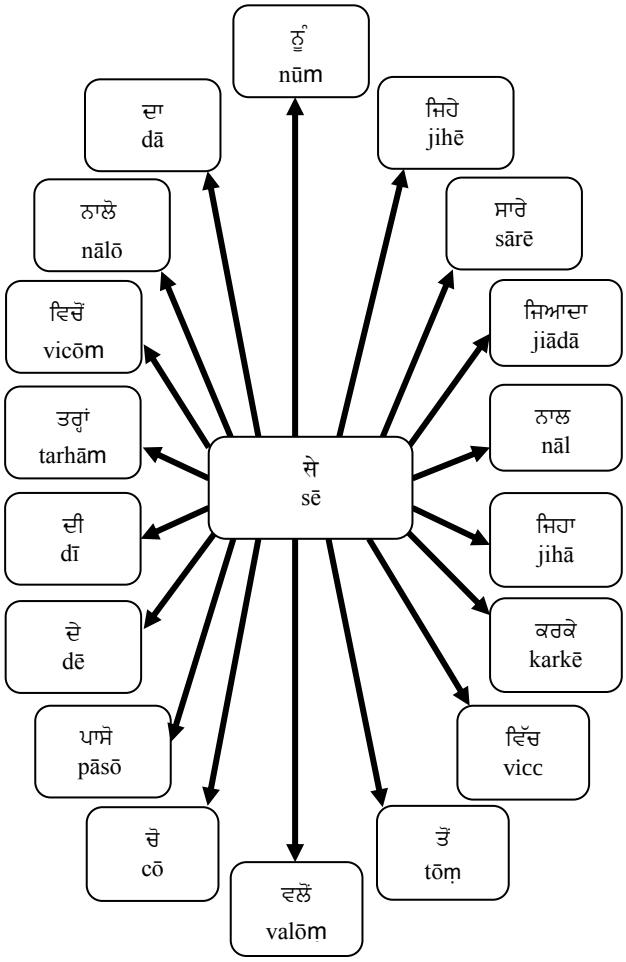

Figure 1: Eighteen possible translations of Hindi post position से (sē) into Punjabi Language

N-gram approach has been chosen to disambiguate post position से (sē) because it does not require sense tagged data which is not available in source and target languages; even though Hindi WordNet is available, developed by IIT Bombay [4], however the post position से (sē) has not been handled in this WordNet.

In this paper, the related work has been discussed in Section 2 followed by a brief description of the existing Punjabi University MT system and the proposed system in Sections 3 and 4 respectively. N-gram algorithm and its smoothing variants are explained in Section 5. Section 6 elaborates methodology of research whereas Section 7 specifies the evaluation metrics. Finally, an analysis of the results of the N-gram system and of the upgraded MT system has been discussed.

2. **Related** work

Bar-Hillel (1964) [5] first raised WSD as a part of Machine Translation (MT). He pointed out that without a "Universal Encyclopedia", a machine would never be able to distinguish between the many meanings of a word. Brown and Pietra (1992) [6] used a much larger text and gave an upper bound of 1.75 bits per character for English language by using trigram (n=3) model. Brown, Della Pietra S.A, Della Pietra V.J., and Mercer (1993) [7] developed five statistical models of translating (sentence to sentence) from one language to another and showed that it is possible to estimate their parameters automatically from a set of pairs of sentences. It was also shown that it is possible to align the words within pairs of sentences algorithmically. Their work mainly was related to English-French machine translation. However their algorithms had minimal linguistic content therefore these can also work well on other pairs of languages. Iyer, Ostendorf and Meteer (1997) [8] investigated the prediction of speech recognition performance, again using trigram model. Chen and Goodman (1998) [9] presented empirical study of smoothing techniques for language modeling. Kilgarriff (1998) [10] defines in his work 17 systems for WSD (supervised and non supervised) that have been tested on SENSEVAL-I. Banerjee and Pedersen (2002) [11] presented an adaptation of Lesk's dictionary-based WSD algorithm which makes use of WordNet glosses and tests on English lexical sample from SENSEVAL-2. Molina, Pla and Segarra (2002) [12] presented a corpus-based approach to WSD based on specialized Hidden Markov Model (HMM). Schiehlen (2003) [13] described two classes of approaches used for WSD i.e. shallow and deep approaches. Shallow approaches simply apply statistical methods to the words surrounding the ambiguous word. On the other hand, deep approaches presume a comprehensive knowledge of the ambiguous word. So far shallow approaches have been more successful since the deep approaches require a higher degree of Artificial Intelligence (AI)

than that attained at present. Vasilescu, Langlais, Lapalme (2004) [14] carried out a series of experiments on the Lesk algorithm, adapted to WordNet and on some variants. Ramakrishnan, Prithviraj, Deepa, Bhattacharyya and Chakrabarti (2004) [15] introduced the notion of soft word sense disambiguation which states that given a word, the sense disambiguation system should not commit to a particular sense, but rather, to a set of senses which are not necessarily orthogonal or mutually exclusive. The senses of a word are expressed by its WordNet synsets, arranged according to their relevance. Sinha, Kumar, Pande, Kashyap and Bhattacharyya (2004) [16] introduced a system to disambiguate nouns in Hindi using Hindi WordNet developed at IIT Bombay. Johnson and Barnard (2005) [17] attempted to disambiguate words using a knowledge base like WordNet and then finding the appropriate sense based on environment information obtained using vision. Josan and Lehal (2008) [2] have attempted to find the optimum value of window size (n) for WSD in Punjabi language. Jurafsky and Martin (2013) [18] used N-gram methods and Maximum Likely-hood Estimates [MLE] to predict the next word after observing preceding (N-1) words. No work has been done for the disambiguation of Hindi post position से (sē) which is the objective of this research.

3. Existing Hindi to Punjabi MT

Hindi and Punjabi are closely related languages having similar structure (SOV). In such languages hybrid approach proves to be simple, easy to implement and an accurate method of MT. The Punjabi University system is based on the hybrid approach for translation of source language to target language. The basic translation approach is the direct word to word translation of Hindi words into Punjabi. After that some rule based approaches were incorporated to remove some obvious glitches. Initially a corpus of direct word to word translation of Hindi words into Punjabi was made. To supplement this, additional modules were created for the various objectives (identifying titles, surnames, Word Sense Disambiguation (WSD), handling out of vocabulary words using transliteration) which makes it as a hybrid system [1,19]. Using this system, a translation success rate of 87.60% was achieved. Most of the prominent ambiguous words were disambiguated, however ambiguity arising from the translation of Hindi post position से (sē) could not be resolved since it has a large number of senses in Punjabi, resolving which is a complete task in itself. The overall translation success rate in respect of this post position से (sē) is only about 2%.

4. Proposed System

The proposed system takes input from the existing Hindi-to-Punjabi machine translation system. N-gram approach and its smoothing variants are then applied to disambiguate post position से (sē) (see Figure 2). These approaches use unigram, bigram and trigram to calculate the joint probability of 18 different senses of से (sē). The results have been analysed manually and appropriate method chosen to integrate with existing system for correct translation of post position से (sē).

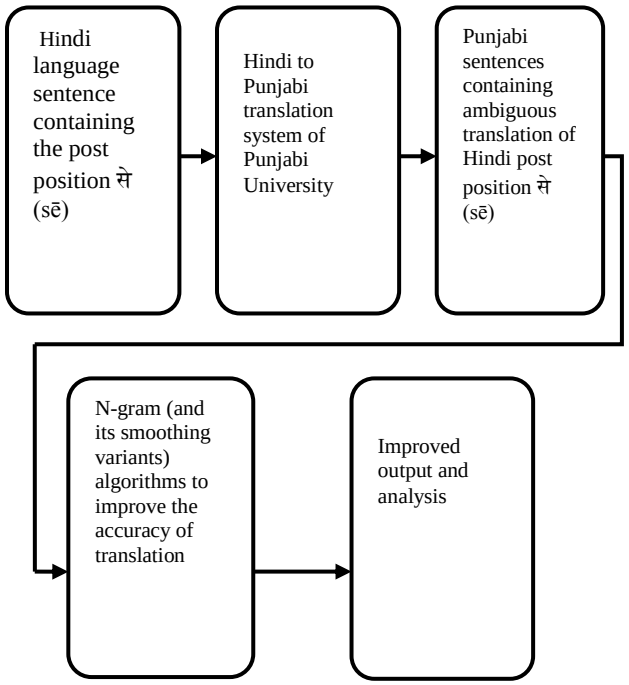

Figure 2. Flow chart of processes followed for analysis

5. N-gram Method

N-gram is a word prediction algorithm using probabilistic methods to predict next word after observing n-1 preceding words. It is simply a sequence of n words along with their count i.e. number of occurrences in training data. Markov assumptions are applied which state that current word does not depend on the entire history of the word but at most on the last few words. Bigram method assumes that we can predict the probability of the alternatives by only looking at the one previous word encountered. It can be generalized to trigram by looking at the prior two words and to N-gram by looking at the past n-1 words. Thus the general equation for the conditional probability of the next word in a sequence would be [18].

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

$$where\ word\ sequence \begin{bmatrix} w_{1,} w_{2,} \dots, w_{n-1} \\ is\ represented\ as\ w_1^{n-1} \end{bmatrix}$$

5.1 Various Estimation Techniques

N-gram methods have a limitation that they have to be trained from some given corpus. Since these training corpus are finite, a new N-gram will erroneously give a zero probability. In addition, MLE method would produce poor estimates when the counts are very small. This limitation can be overcome by assigning some non-zero values to these low probability N-grams [19, 20].

5.1.1 Add One Smoothing

In this we add one to all the counts (frequencies) before normalizing them into probabilities.

$$Bigram\ P(w_n \mid w_{n-1}) = \frac{c(w_{n-1}\, w_n) + 1}{c(w_{n-1}) + V} \quad (2)$$

$$Trigram\ P(w_n \mid w_{n-1}\, w_{n-2}) = \frac{c(w_{n-1}\, w_{n-2}\, w_n) + 1}{c(w_{n-1}\, w_{n-2}) + V} \quad (3)$$

$$NGram\ P(w_n \mid w_{n-N+1}^{n-1}) = \frac{c(\, w_{n-N+1}^{n-1}\ \ w_n) + 1}{c(w_{n-N+1}^{n-1}) + V} \quad (4)$$

Where V is the total number of possible (n-1) grams vocabulary size [18].

This shows improved results (see Table 4) as compared to simple bigram and trigram methods.

5.1.2 Backoff Smoothing

Backoff N-grams modeling is a nonlinear method introduced by Katz(1987). Backoff smoothing uses all preprocessing steps defined in the previous methods. It uses all three approaches i.e. Trigram, Bigram and Unigram. In this approach, first, all the combinations of trigrams related to the ambiguous words are created and probability of all the trigram data calculated. The sense with highest probability is taken as the correct alternative. If no substitute of the trigram is found in the training data, then system goes back to the bigram and if again unsuccessful then it goes to the unigram [18, 15].

$$P(w_n \mid w_{n-2}\, w_{n-1}) = \begin{cases} P(w_n \mid w_{n-2}\, w_{n-1}), & if\ c(w_{n-2}\, w_{n-1}\, w_n) > 0 \\ P(w_n \mid w_{n-1}), & if\ c(\ w_{n-1}\, w_n) > 0 \\ P(w_n), & otherwise \end{cases} \quad (5)$$

5.1.3 Deleted Interpolation

Deleted Interpolation is alternative to the Backoff algorithm, where information is used from the lower N-grams even if the higher order N-gram count is non zero. The deleted interpolation algorithm, due to Jelinek and Mercer (1980), combines different N-gram orders by linearly interpolating all three models whenever there is computation of any trigram. That is to estimate the probability by mixing together the unigram, bigram and trigram probabilities. Each of these is weighted by a linear weight λ [18].

$$P(w_n \mid w_{n-1}\, w_{n-2}) = \lambda_1 P(w_n \mid w_{n-1}\, w_{n-2}) + \lambda_2 P(w_n \mid w_{n-1}) + \lambda_3 P(w_n)$$

$$\sum_{i=1}^{i=3} \lambda_i = 1 \quad (6)$$

The values of individual λ_i were determined by hit and trial method i.e. $\lambda_1 = 0.6,\ \lambda_2 = 0.3,\ \lambda_3 = 0.1$

5.1.4 Witten-Bell Discounting

Witten-Bell discounting is based on zero-frequency events. Lets think of a zero-frequency word or N-gram as one that just has happened yet. When it first

occurs, its probability will be as follows [18].

$$P(w_i \mid w_{i-1}) = \frac{T(w_{i-1})}{z(w_{i-1})(N+T(w_{i-1}))} \quad if\ (w_{i-1}\ w_i) = 0 \quad (7)$$

$$P(w_i \mid w_x) = \frac{C(w_x\ w_i)}{C(w_x)+T(w_x)} \quad if(w_{i-1}\ w_i) \neq 0 \quad (8)$$

where T=total number of words starts with previous word or

bigram type.

Z= total words- words start with previous word.

N= number of bigram tokens on the previous word.

6. Methodology of Research

First, Punjabi corpus containing about 1.7 million words wascollected from various sources i.e. newspapers, books, TDIL (Technology Development for Indian Languages), Punjabi University and IIIT Hyderabad. This text was then manually normalized by removing all unnecessary text, alphanumeric characters and symbols. Next, trained data was created as unigrams, bigrams and trigrams and frequency of each was calculated from the raw corpus. This turned out to be 745139 unigrams, 745138 bigrams and 689835 trigrams (with combination of eighteen different senses in Punjabi) in the database to get correct translation of Hindi post position से (sē) while translating into Punjabi.

Excerpts from training corpus as unigram, bigram and trigram with their respective frequencies are shown in Table 1, Table 2 and Table 3 respectively. These frequencies of words were calculated from the Punjabi raw corpus already collected. These frequencies are used to find the conditional probability of all combinations of post position से (sē). Then N-gram algorithms and its smoothing variants select the highest probability for the translation of post position से (sē) in the given context. In other words, the entries of these tables are used as database of our system to get correct translation of the post position से (sē).

Table 1. Unigram database

Unigram	Roman Script of Unigram for Punjabi Word	Frequency
ਦੇ	dē	16459
ਜਦੋਂ	jadōṃ	9543
ਸਾਥੀਆਂ	sāthīāṃ	373
ਕੋਰਟ	kōraṭ	1301
ਨੇ	nē	21232
ਸਿੱਕੀਮ	sikkīm	341
ਅੱਜ	ajj	561
ਕਰਕੇ	karkē	407

Table 2. Bigram database

Bigram	Roman Script of Bigram for Punjabi Word	Frequency
ਅੱਜ ਸਵੇਰੇ	ajj savērē	2032
ਭੀੜ ਚੋ	bhīṛ cō	248
ਕੋਰਟ ਵਿੱਚ	kōraṭ vicc	1234
ਪੇਸ਼ ਕੀਤਾ	pēsh kītā	1017
ਗਿਆ ਤਾਂ	giā tāṃ	1591
ਕੋਰਟ ਦੇ	kōraṭ dē	984
ਦੇਣ ਕਰਕੇ	dēṇ karkē	107

Table 3. Trigram database

Trigram	Roman Script of Trigram for Punjabi Word	Frequency
ਅੱਜ ਸਵੇਰੇ **ਤੋਂ**	ajj savērē tōṃ	539
ਸਾਥੀ ਜਬਲਪੁਰ **ਵਲੋਂ**	sāthī jablapur valōṃ	323
ਦੰਗੇ ਪੀੜੀਤਾਂ **ਨਾਲ**	daṅgē pīḍaītāṃ nāl	432
ਛੱਡ ਦੇਣ **ਕਰਕੇ**	chaḍḍ dēṇ karkē	284
ਵਿਅਕਤੀ ਠੀਕ **ਤਰ੍ਹਾਂ**	viaktī ṭhīk tarhāṃ	192
ਦਾਖਲ ਹੋਣ **ਕਰਕੇ**	dākhal hōṇ karkē	387
ਦੇ ਪੈਮਾਨੇ **ਤੇ**	dē paimānē tē	170
ਦੇ ਪਿੱਛੇ **ਦੀ**	dē picchē dī	185

Steps of Algorithm (For Bigram Approach)

Step1: Original Hindi language sentence to be translated into Punjabi language

कुल्लू के सभी लोग इस देवते को एस.पी. के नाम से

ਪੁਕਾਰਤੇ ਹੈਂ।
kullū kē sabhī lōg is dēvtē kō ēs.pī. kē nām sē pukārtē haiṃ.

Step2: Inaccurate Punjabi translation performed by the exiting Hindi-to-Punjabi machine translation system of Punjabi University (i.e. before updation).
ਕੁੱਲੂ ਦੇ ਸਾਰੇ ਲੋਕ ਇਸ ਦੇਵਤੇ ਨੂੰ ਏਸ . ਪੀ . ਦੇ ਨਾਮ ਵਲੋਂ ਬੁਲਾਉਂਦੇ ਹਨ।
kullū dē sārē lōk is dēvtē nūṃ ēs pī dē nām valōṃ bulāundē han

Step3: To normalize Punjabi sentence by removing all unwanted symbols (like #, |, @, ! etc.)
ਕੁੱਲੂ ਦੇ ਸਾਰੇ ਲੋਕ ਇਸ ਦੇਵਤੇ ਨੂੰ ਏਸ ਪੀ ਦੇ ਨਾਮ ਵਲੋਂ ਬੁਲਾਉਂਦੇ ਹਨ
kullū dē sārē lōk is dēvtē nūṃ ēs pī dē nām valōṃ bulāundē han

Step 4: Tokenization to separate the words of string and store it into the linked list
{<Start>}-> { ਕੁੱਲੂ }-> { ਦੇ }-> { ਸਾਰੇ }-> { ਲੋਕ }-> { ਇਸ }-> { ਦੇਵਤੇ }-> { ਨੂੰ }-> { ਏਸ }->{ ਪੀ }-> { ਦੇ }->{ ਨਾਮ } -> { ਵਲੋਂ }-> { ਬੁਲਾਉਂਦੇ }-> { ਹਨ ->{<Stop>}

Step 5: Give this tokenized input to Bigram Function to check for ambiguous word ਵਲੋਂ (valōṃ).

Step 6: If it exists, bigram function will return appropriate translation using highest joint probability of eighteen different substitutions of word ਵਲੋਂ (valōṃ) like Joint probability of ਕੋਰਟ ਵਿੱਚ is $P(\text{ਵਿੱਚ} \mid \text{ਕੋਰਟ}) = \frac{c(\text{ਕੋਰਟ ਵਿੱਚ})}{c(\text{ਕੋਰਟ})} = \frac{1234}{1301} = 0.94$

Steps of Algorithm (For Trigram Approach)

In trigram algorithm all preprocessing of text is same as given in the bigram. In this algorithm the frequency of trigram and bigram is used to find the joint probability of the given trigrams like ਅੱਜ ਸਵੇਰੇ ਤੋਂ Joint probability of ਅੱਜ ਸਵੇਰੇ **ਤੋਂ** is $P(\text{ਤੋਂ} \mid \text{ਅੱਜ ਸਵੇਰੇ}) = \frac{c(\text{ਅੱਜ ਸਵੇਰੇ ਤੋਂ})}{c(\text{ਅੱਜ ਸਵੇਰੇ})} = \frac{539}{2032} = 0.26$

In the same way the joint probability of all other senses of ਸੇ (sē) is calculated. The highest probability is taken by the algorithm which is considered as the best estimate of the translation and hence replaces ਵਲੋਂ (valōṃ) with this substitute.

Smoothing techniques have also been applied to disambiguate the post position ਸੇ (sē). These techniques follow Steps 1 to 5 defined in bigram approach and Step 6 uses the respective algorithm (refer Section 5) to disambiguate the post position ਸੇ (sē). The results are shown in Table 4 and Table 5.

7. Evaluation Metrics and Test Data

The improved system after disambiguating ਸੇ (sē) has been evaluated using the standard evaluation metrics of information retrieval viz. precision, recall and the combined F-measure metric. Precision (P) tells us how much of the correct information has been returned by the system whereas Recall (R) indicates how much relevant information has been extended by our method. F-measure (F) is a combined metric which balances the recall and precision metrics by giving them appropriate weights [18].

For testing of system, three thousand twenty (3020) sentences were collected which contained Hindi post position ਸੇ (sē).These were taken from newspapers, books, popular magazines etc. covering diverse fields like politics, sports, newspaper articles, short stories, entertainment, science & technology and education. These sentences were put through the Hindi-to-Punjabi machine translation system (http://h2p.learnpunjabi.org) developed at Punjabi, University [1]. The translated Punjabi sentences were collected and put to N-gram (and its smoothing of translation variants) algorithms and the improvement in translation accuracy was analysed (see Figure 2).

8. Results and Analysis

It is seen that algorithms (N-gram and its variants) improved the translation accuracy of Hindi post position ਸੇ (sē) into Punjabi appreciably. The F-measure which was 0.024 in Hindi to Punjabi translation without using any WSD algorithm

Table 4. Analysis using Bigram, Trigram, Bigram with Add-One smoothing algorithms

Text Domain	Size of Domain	Bigram			Trigram			Bigram with Add-One Smoothing		
		Precision (P)	Recall (R)	F-Measure (F)	Precision (P)	Recall (R)	F-Measure (F)	Precision (P)	Recall (R)	F-Measure (F)
Politics	400	0.759894	0.932039	0.837209	0.789474	0.714286	0.75	0.767263	0.970874	0.857143
Sports	500	0.7431	0.923483	0.823529	0.778364	0.707434	0.741206	0.772541	0.969152	0.859749
News Paper Articles	650	0.764992	0.934653	0.841355	0.79065	0.711152	0.748797	0.766929	0.97012	0.85664
Short Stories	370	0.765714	0.930556	0.840125	0.785714	0.709677	0.745763	0.760563	0.947368	0.84375
Entertainment	450	0.741093	0.914956	0.818898	0.787611	0.706349	0.74477	0.770455	0.971347	0.859316
Science & Technology	290	0.735294	0.917431	0.816327	0.785388	0.707819	0.744589	0.741259	0.981481	0.844622
Education	360	0.742604	0.919414	0.821604	0.787546	0.711921	0.747826	0.769231	0.967742	0.857143
Total	3020	0.751756	0.925638	0.829684	0.786527	0.70983	0.746213	0.765445	0.968227	0.854976

Table 5. Analysis using Trigram with Add-One smoothing, Backoff smoothing, Deleted Interpolation, Witten-Bell algorithms

Text Domain	Size of Domain	Trigram with Add-One Smoothing			Backoff Smoothing			Deleted Interpolation			Witten-Bell		
		Precision (P)	Recall (R)	F-Measure (F)	Precision (P)	Recall (R)	F-Measure (F)	Precision (P)	Recall (R)	F-Measure (F)	Precision (P)	Recall (R)	F-Measure (F)
Politics	400	0.664103	0.962825	0.786039	0.792627	0.484507	0.601399	0.793578	0.487324	0.603839	0.644351	0.488889	0.555957
Sports	500	0.667347	0.970326	0.79081	0.785185	0.479638	0.595506	0.789668	0.481982	0.598601	0.736842	0.455814	0.563218
News Paper Articles	650	0.665615	0.96347	0.787313	0.787535	0.483478	0.599138	0.792614	0.483536	0.600646	0.759531	0.455986	0.569857
Short Stories	370	0.674033	0.968254	0.794788	0.792079	0.487805	0.603774	0.792079	0.484848	0.601504	0.61435	0.482394	0.540434
Entertainment	450	0.668182	0.967105	0.790323	0.783133	0.492424	0.604651	0.795082	0.485	0.602484	0.724576	0.444156	0.550725
Science & Technology	290	0.664311	0.964103	0.786611	0.78481	0.484375	0.599034	0.788462	0.478599	0.595642	0.691358	0.466667	0.557214
Education	360	0.666667	0.962963	0.787879	0.784615	0.481132	0.596491	0.794872	0.484375	0.601942	0.676617	0.461017	0.548387
Total	3020	0.667119	0.965653	0.789094	0.787105	0.484644	0.599907	0.79243	0.483787	0.600787	0.698441	0.462853	0.55675

improved to 0.829 using bigram method and rose to 0.8549 for bigram with Add-One smoothing which gave the highest accuracy amongst all our algorithms (see Table 4 and Table 5).

Any appreciable difference in success rate in Precision (P), Recall (R) and F-Measure (F) metrics among the various domains like politics, literature, sports etc. was not noticed (see Table 4 and Table 5). It also shows that our methods used (to improve the translation success rate) are domain independent (which is a desirable feature). Tables 4 to 5 also show the P, R and F values for whole sample size (taking all domains together) which is found to be similar to our results in the different domains.

This n-gram method was finally integrated with the existing Punjabi University system. We will call this integrated system as the upgraded Hindi to Punjabi machine translation system. Ten persons each belonging to a different profession were selected for testing of the upgraded system. Each one of them was asked to collect three hundred Hindi sentences from any source (books, newspapers etc.) and input these into the upgraded translation system. The Punjabi translated output was manually analysed on a score of 0 to 3 (BLEU score) with 3 as perfectly clear and intelligible, 2 as generally clear and intelligible, 1 as hard to understand while 0 score account that the translated sentences were not understandable at all.

The results were as follows.

- 71.3% of translated sentences obtained a score of 3
- 26.01% of translated sentences obtained a score of 2
- 2.03% of translated sentences obtained a score of 1
- 0.66% of translated sentences obtained a score of 0

Therefore, we concluded that a total of 97.31% translated sentences were intelligible. Hence the accuracy of translation has improved from 87.60% to 92.30% (according to BLEU score).

9. Conclusions and Scope for Future Work

Performance of translation of Hindi post position से (sē) can be improved considerably by using various statistical algorithms like N-gram methods along with its smoothing variants. It has been seen that out of eighteen possible translations of this Hindi post position से (sē) to Punjabi, the translation accuracy improved from 2% to 85.49% when these WSD methods were applied. Best results were obtained when bigram method was applied with Add-One smoothing (see Table 4). Those results were obtained using the testing corpus of 3020 Hindi language sentences containing word से (sē)**.** Better results are likely to be achieved with a large trained corpus and more sophisticated smoothing algorithms. Hence the overall accuracy of translation using upgraded system has been improved from 87.60% to 92.30%.

References

[1] Goyal, V., Lehal, G.S. (2011). Hindi to Punjabi Machine Translation System. *Proceeding of the ACL-HLT System Demonstrations. Portland, Oregon, USA.* pp. 1-6

[2] Josan, G. S., and Lehal, G. S. (2008). Size of N for Word Sense Disambiguation using N Gram Model for Punjabi Language, *International Journal of Translation*, Vol. 20, No. 1-2, pp. 47-56.

[3] Goyal, V., and Lehal, G.S. (2011). N-gram Based Word Sense Disambiguation: A Case Study of Hindi to Punjabi Machine Translation system. *International Journal of Translation*, Vol. 23, No. 1, pp. 99-113.

[4] Narayan, D., Chakrabarty, D., Pande, P., Bhattacharyya, P. (2002). An Experience in Building the Indo WordNet- a WordNet for Hindi, *International Conference on Global WordNet (GWC 02), Mysore, India*.

[5] Bar-Hillel, Y. (1964). On Syntactic categories, *Journal of Symbolic Logic 15.1-16* [Repr.Bar-Hillel 1964a, 19-37].

[6] Brown, P.F., Pietra, S.A.D. (1992). An

Estimation of Upper Bound for the Entropy of English, *Association for Computational Linguistics, Volume 18, Number 1,* pp. 31-40.

[7] Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L. (1993). The Mathematics of Statistical Machine Translation: Parameter estimation. *Computational linguistics 19(2), 263–311.*

[8] Iyer, R., Ostendorf, M., Meteer, M. (1997). Analyzing and Predicting Language Model Improvements, *In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understandin.*

[9] Chen, S. F., Goodman, J. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. *Proceedings of the 34th Annual Meeting of ACL*

[10] Kilgarriff, A. (1998). SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation, *Programs In Proc. LREC, Granada.* pp. 581-588.

[11] Banerjee, S., Pedersen, T. (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. *CICLing, Mexico*.

[12] Molina, A., Pla, F., Segarra, E. (2002). A Hidden Markov Model Approach to Word Sense Disambiguation, *In Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence IBERAMIA, London, UK*. pp. 655-66.

[13] Schiehlen, M. **(**2003). Combining Deep and Shallow Approaches in Parsing German. *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp. 112-119.

[14] Vasilescu, F., Langlais, P., Lapalme, G. (2004). Evaluating Variants of the Lesk Approach for Disambiguating Words. *LREC, Portugal*.

[15] Ramakrishnan, G., Prithviraj, B.P., Deepa, A., Bhattacharyya, P., Chakrabarti S. (2004). Soft Word Sense Disambiguation. *International Conference on Global WordNet (GWC 04), Brno, Czeck Republic.*

[16] Sinha, M., Kumar, M., Pande, P., Kashyap, L., Bhattacharyya, P. (2004). Hindi Word Sense Disambiguation, *International Symposium on Machine Translation, Natural Language Processing and Translation Support Systems, Delhi, India*.

[17] Johnson, M., Barnard, K. (2005). Word Sense Disambiguation with Pictures.

[18] Jurafsky, D., Martin, J.H. (2013). Speech and Language Processing. *Pearson, Eighteenth Impression.*

[19] Goyal, V., Lehal, G.S. (2009). Evaluation Hindi to Punjabi Machine Translation System. *International Journal of Computer Science Issues, vol. 4, No. 1, ISSN 1694-0814*

[20] Zhai, C., Lafferty, J. (2001). A Study of Smoothing Methods for Language Model Applied to ad hoc Information Retrieval. *SIGIR 01*, New Orleans, Louisiana, USA, pp 9-12