LANGUAGE LEARNING AND TRANSLATION WITH UBIQUITOUS APPLICATION THROUGH STATISTICAL MACHINE TRANSLATION APPROACH

Sandeep R. Warhade[1], Prakash R. Devale[2] and S. H. Patil [3]
[1]Research Scholar, Deptt. of IT,Bharati Vidyapeeth Deemed University College of Engineering, Pune-India.
[2]Professor & Head, IT Deptt.,Bharati Vidyapeeth Deemed University College of Engineering, Pune-India.
[3]Professor & Head, Comp. Deptt.,Bharati Vidyapeeth Deemed University College of Engineering, Pune-India.

ABSTRACT

This paper describes the Phrase-Based Statistical Machine Translation Decoder for English to Sanskrit translation in ubiquitous environment. Our goal is to improve the translation quality by enhancing the translation table and by preprocessing the Sanskrit language text . We introduce a comprehensive framework for a ubiquitous translation and language learning environment utilizing the capabilities of modern cell phone technology. We present the architecture of our framework, our current state of implementation, and the findings we have gathered so far.

KEYWORDS: *English Sanskrit Translation, Statistical Machine Translation, Ubiquitous Translation, Ubiquitous Computing.*

I. INTRODUCTION

Automatic translation from one natural language into other using computers is the Machine Translation. Statistical machine Translation is an approach to MT that is characterized by the use of machine learning methods. This means that we apply a learning algorithm to large body of previously translated text, known variously as a parallel corpus, parallel text, bi-text or multi-text. With an SMT toolkit and enough enough parallel text, we can build an MT system for a new language pair within a very short period of time. The accuracy of these systems depends crucially on the quantity, quality and domain of the data. Making a sequence of word translation and reordering decisions perform translation. Word translation is often ambiguous, means it is common for the different possible translations of a word to have very different meanings. Often the correct choice will depend on context. Therefore, our system will need some mechanism to correctly reorder the words. Reordering is typically dependent on the systematic structure of the target language. As with word translation, reordering decisions often entail resolving some kind of ambiguity. English is a well known language so we illustrate Sanskrit grammar and its salient features. The English sentence always has an order of Subject-Verb-Object, while Sanskrit sentence has a free world order. A free order language is natural language which does not lead to any absurdity or ambiguity thereby maintaining a grammatical and semantic meaning for every sentence obtained by the change in the ordering of the words in the original sentence. For example, the order of English Sentence (ES) and itself equivalent translation in Sanskrit Sentence (SS) is given as below.

ES:	Ram	reads	book.
	(Subject)	(Verb)	(Object)
SS:	Raamah	pustkam	pathati.
	(Subject)	(Object)	(Verb)
	Pustkam	raamah	pathati.
	(Object)	(Subject)	(Verb)
	Pathati	pustkam	raamaha.
	(Verb)	(Object)	(Subject)

Thus Sanskrit sentence can be written using SVO, SOV and VOS order.

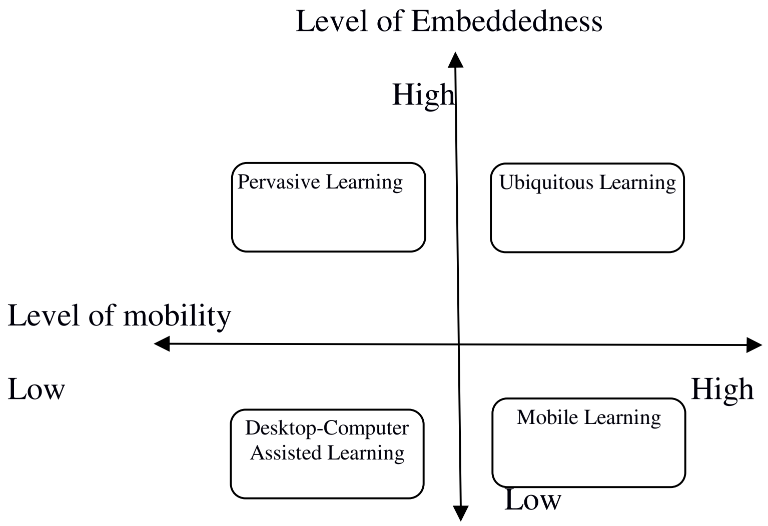

Fig. 1: Types of Learning Environment [5]

According to [5], Fig. 1 demonstrates the correlation of the important terms in a computer-assisted learning context. Computer assisted learning is usually software or Web content accessible from a local workstation. The easier it is for the students to carry around this program and access it wherever they are, e.g. utilizing a PDA, the more it becomes a mobile learning application. The pervasive component is added if the application and the device are also able to measure and/or adjust to the environment of the students. A combination of pervasive and mobile learning is called ubiquitous learning and is defined by a high degree of embeddedness and mobility. Similarly, we define a ubiquitous translation system as a mobile translation system, which uses information about the current situation to refine or expand the translation, make suggestions for useful terms, sentences, names, etc.

In order to transform this theory into a real application, a proper platform is needed: a device which offers portability, computational power, sensory capabilities, and ease of use. The modern cell phone meets all those prerequisites and is, beyond that, already widespread. As Mark Weiser puts it:

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [11]

II. LIKE WORK HISTORY

To build a functioning SMT system, there are four problems that we must solve

1) Translational Equivalence model or models: Transformation of source sentence into a target sentence have series of steps. These series of steps are called a translation equivalence model or simply a model.

2) Parameterization: To resolve some ambiguity we want to enable our model to make good choices when faced with a decision. Parameterization of a model will enable us to assign a score to every possible source and target sentence pair that our model might consider.

3) Parameter estimation: The parameterization defines a set of statistics called parameters used to score the model, but we need to associate values of these parameters. This is called parameter estimation.

4) Decoding: With the sentence presented, we must search highest-scoring translation according to our model. This is called decoding.

There is a variety of Web-based translation services and language learning applications available, e.g. [2, 4, 8]. The translation services adopt techniques from machine translation, which has ever since included many different approaches. An overview of those can be obtained from [6]. Based on [4, 17, 7] we have found that our method, being example and corpus-based, is well suited for a language learning application, since it contains intermediate information in the translation processes, which can be valuable for a language student. The ubiquitous property of mobile devices has been successfully used to offer contextual learning environments in the past [13, 3, 10]. One of the implementations is a system by [12], where the learners could utilize PDAs to find the right politeness form while formulating a Japanese sentence in a certain situation. The application of ubiquitous learning demands several characteristics to be fulfilled [9]:

- Permanency: Learners never lose their work unless it is purposefully deleted. In addition, the entire learning process is recorded continuously.

- Accessibility: Learners have access to their documents, data, or videos from anywhere. That information is provided based on their requests. Therefore, the learning involved is self-directed.

- Immediacy: Wherever learners are, they can get any information immediately. Thus, learners can solve problems quickly. Otherwise, the learner can record the question and look for the answer later.

- Interactivity: Learners can interact with experts, teachers, or peers in the form of synchronous or asynchronous communication. Hence, the experts are more reachable and the knowledge becomes more available.

- Situation of instructional activities: The learning could be embedded in our daily life. The problems encountered as well as the knowledge required are all presented in their natural and authentic forms. This helps learners to notice the features of problem situations that make particular actions relevant.

An example of the application of those guidelines can be seen in [1]. The idea behind it is to give the students the chance to efficiently use their time and the ability to access class room information at will. In [14] a ubiquitous learning environment was developed by using IEEE 802.11 WLAN and Bluetooth for network communication. This showed that a learning experience, supported by a contextually matching surrounding, is more valuable in terms of understanding and memorizing since it is based on an inductive process. However, the limits of the network technologies do not allow a deployment of that system into a large network structure.

III. SYSTEM FRAMEWORK

The framework is designed as a client-server setup. The client is the development board, in our case the Android Emulator. The LAMP (Linux-Apache-MySQL-PHP) server is situated at the Bhartee Vidyapeeth College of Engineering, Pune. The hardware specifications of both are listed in Table 1.

Table 1 : Hardware Configuration

	Server	Client -
CPU	4x Intel(R) Xeon(R) CPU E5405 @ 2.00GHz	Android Emulator
RAM	4057MB	64 MB
OS	Ubuntu 9.10	Android
Sec. Mem.	1.7 TB	1 GB

The operating system on the development board is Android, the open-source, running on a Linux kernel, designed for smart phones and Internet tablets. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. It is developed by the Open Handset Alliance™ , led by Google, and other companies. The Open Handset Alliance™ is a group of mobile operators, hardware manufacturers, semiconductor companies, software companies and commercialization companies. For development we have used Eclipse, a editor, emulator and cross-compilation toolkit, in combination with Java. In contrast to distributions for desktop computers, Android has touch screen support, sliding keyboard support, an interface to the Camera, GPS, compass, and accelerometer, while discarding some typical desktop distribution functionalities. Even though the hardware on the client side is quite powerful for a cell phone, it is not enough to perform all the calculations needed for our framework in a reasonable amount of time. Hence, we need to outsource as many calculations as possible to the server, where they can be processed much quicker. We have considered the fact that a cellular phone is not always guaranteed to have a decent network connection. Spots without a carrier signal, such as tunnels, elevators, etc. have to be taken into consideration. Therefore, the communication between the server and the client is done over asynchronous calls, to guarantee a service even in the case of an interruption of the network connection. Additionally, a database on the mobile device is kept to store user input, such as vocabulary lists or program preferences, and enable system use while the network and/or the server is not available. Outdated entries from this database are purged periodically, due to limited storage capabilities on the cell phone. Fig. 2. The translation module presents the contents graphically and provides input fields. The language learning module offers a graphical learning program with various functionality. Data from the knowledge base on the server are used to construct a customized learning support, in terms of word/sentence suggestions and difficulty level. Both inter faces store essential data on the device. The language learning module accesses the personal database on the phone, which holds the necessary information to operate the framework without a network connection, though with limited capabilities.

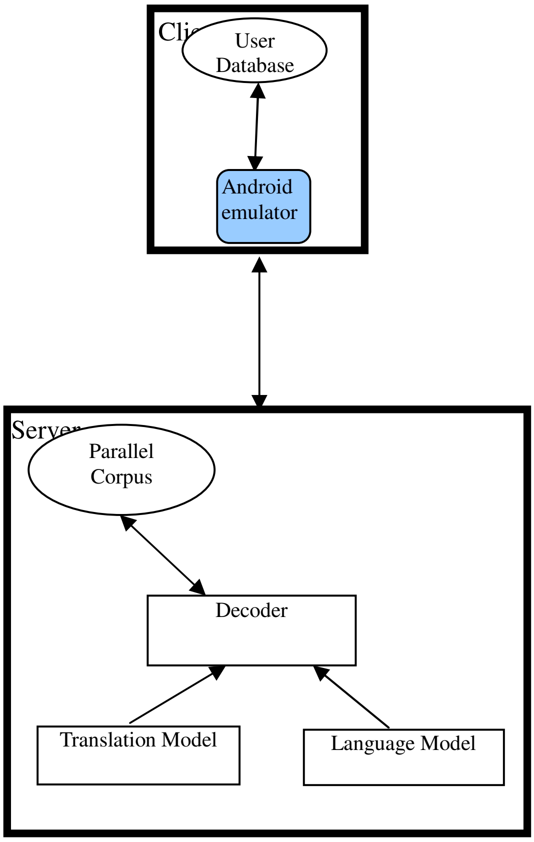

Fig. 2: System Architecture

This personal database is synchronized with the user-specific database on the server, whenever possible. The user-specific database, stores the user's history, such as previous query sentences. Words, compounds, and sentences stored in this database are assumed to be of interest to the student and are preferred in lecture suggestions.

IV. Showcase

As a translation basis we take the output of the statistical machine translation system Moses [4]. The overview of the data flow in the system is shown in Fig. 3. An input sentence is sent to the part-of-speech (PoS) tagger for English and Sanskrit as translation model. After each sentence token is assigned a PoS-tag, the sentence and its tags are compared with sentences from a preformatted corpus. For this purpose, we have modified and enriched a bilingual data collection consisting of 1500 sentences, taken from Sanskrit learning books.

We have removed as much noise as possible from the data, assigned PoS-tags to each sentence token and stored the information in an SQL database. We have created different formats of the bilingual data, one with a complete set of PoS information and others with reduced and optimized tag sets to provide quick access and efficient processing. Additional representations and tag sets can be added easily to satisfy different needs in future work. We have applied relational sequence alignment [16] to obtain clusters of structurally similar sentences, so that the comparison of the query sentence with the clusters yields several similar structures. At the same time, the query sentence is processed with Moses to obtain a preliminary translation. This translation is then used to fill the template of the structures, which had been found to be similar in terms of PoS-tags. This way, a certain number of translation candidates is produced. The parameters of the similarity measure can be adjusted to fine-tune the result, depending on the text type and text domain. Allowing low threshold values for similarity, a higher number of candidates can be produced, whereas a higher threshold value reduces the number of candidates.

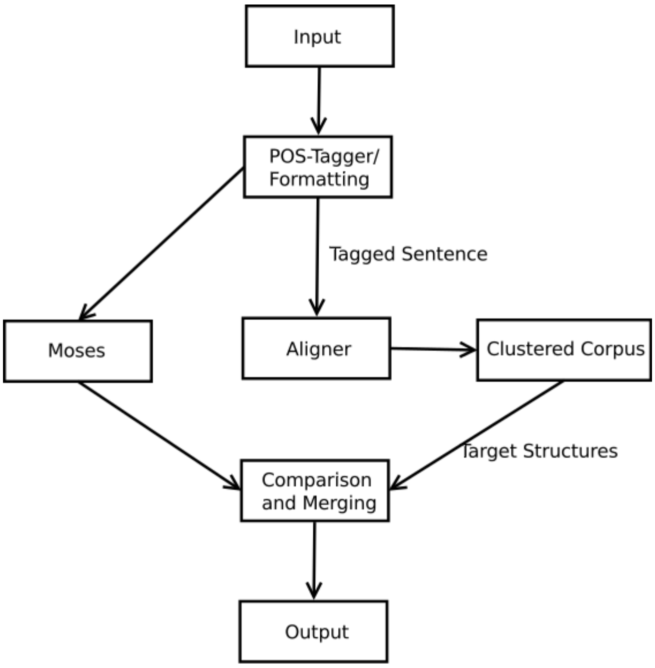

Fig. 3: Data Flow of System.

The Translation Task from the Starting View initiates work flow as shown in Fig. 4.

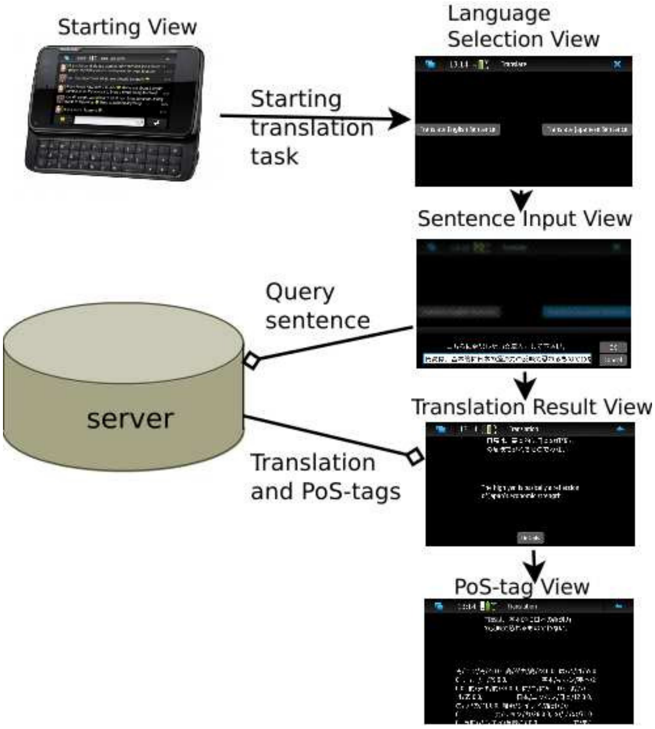

Fig. 4: Work Flow of Translation Task

Upon initiating the task, a connection to the server is attempted. In the first view, the user has the choice of the input language. An input box takes the query sentence and sends it to the decoder module, which is located at the server. In the next view, the translation result is displayed and the query sentence is stored on the server for later use by the didactic module. Additionally, the user has the choice of viewing further language details. At the time of writing, we display various PoS-tag information. In the future we plan to integrate a more detailed linguistic analysis, e.g. dependency trees produced by CaboCha [18].

V. CONCLUSIONS

In this paper we have described the design and a design of a ubiquitous translation and language learning framework, in particular for English to Sanskrit, on the android emulator, a growing cellular phone operating system with internet capabilities. We have presented our implementation of a translation and language learning environment built as a client-server system, which consists of a translation and a language learning task. For the language learning task we have built a learning application. For the translation task we used statistical machine decoder, a translation framework. By integrating SMT decoder into this research work, we have shown how a client/server configuration can be realized to offer the entire translation service on the mobile device. In future work, we want to focus on audio input, GPS localization, and user query statistics to create a detailed user profile. This will facilitate a specific fine-tuning of the learning environment, considering guidelines for computer enhanced learning such as the cognitive load theory, and the GUI design. As mentioned before, the already implemented learning environment is available on the Web to receive feedback from the Android community. In addition, we will make our framework available to students at our Language Department, once it is in a workable state. We will evaluate the framework with a sufficient number of users and a control group along the dimensions engagement, effectiveness, and viability [15], as well as analyze the usability on other prevalent mobile platforms, such as iPhone, Maemo5's successor MeeGo, etc.

ACKNOWLEDGEMENTS

The authors would like to thank everyone, just everyone!

REFERENCES

[1]. B. Bomsdorf. Adaptation of learning spaces: Supporting ubiquitous learning in higher distance education. In N. Davies, T. Kirste, and H. Schumann, editors, Mobile Computing and Ambient Intelligence: The Challenge of Multimedia, number 05181 in Dagstuhl Seminar, Dagstuhl, Germany, 2005.

[2]. C. Goutte et al., editors. Learning Machine Translation. MIT Press, Cambridge, Massachusetts, 2009.

[3]. H. Ogata et al. Computer supported ubiquitous learning environment for Japanese mimicry and onomatopoeia with sensors. In Proceedings of the 2007 Conference on Supporting Learning Flow through Integrative Technologies, pages 463–470, Amsterdam, The Netherlands, 2007. IOS Press.

[4]. W. Winiwarter. WILLIE – a Web Interface for a Language Learning and Instruction Environment. In Proceedings of the 6th International Conference on Web-based Learning, Edinburgh, United Kingdom, 2008. Springer-Verlag.

[5]. K. Lyytinen and Y. Yoo. Issues and challenges in ubiquitous computing. Commun. ACM, 45(12):62–65, 2002.

[6]. Y. Wilks. Machine Translation: Its Scope and Limits. Springer-Verlag, 2008.

[7]. W. Winiwarter. WETCAT – Web-Enabled Translation using Corpus-based Acquisition of Transfer rules. In Proceedings of the Third IEEE International Conference on Innovations in Information Technology, Dubai, United Arab Emirates, 2006.

[8]. N. Nagata. Banzai: Computer assisted sentence production practice with intelligent feedback. In Proceedings of the Third International Conference on Computer Assisted Systems for Teaching and Learning/Japanese (CASTEL/J), 2002.

[9]. Y. Chen et al. A mobile scaffolding-aid-base bird-watching learning system. In Proceedings of the IEEE International Workshop on Wireless and Mobile Technologies in Eduction, pages 15–22. IEEE Computer Society Press, 2002.

[10]. L. H. Gan et al. Language learning outside the classroom using handhelds with knowledge management. In Proceedings of the 2007 Conference on Supporting Learning Flow through Integrative Technologies, pages 361–368, Amsterdam, The Netherlands, 2007. IOS Press.

[11]. M. Weiser. The computer for the 21st century. SIGMOBILE Mob. Comput. Commun. Rev., 3(3):3–11, 1999.

[12]. H. Ogata and Y. Yano. Context-aware support for computer-supported ubiquitous learning. In Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04), page 27, Washington, DC, USA, 2004. IEEE Computer Society.

[13]. C. Yin, H.Ogata, and Y. Yano. JAPELAS: Supporting Japanese polite expressions learning using PDA(s) towards ubiquitous learning. International Journal of Information and Systems in Education, 3(1):33–39, 2005.

[14]. V. Jones and J. H. Jo. Ubiquitous learning environment: An adaptive teaching system using ubiquitous technology. In Proceedings of the 21st ASCILITE Conference, 2004.

[15]. D. A. Norman and J. C. Spohrer. Learner-centered education. Commun. ACM, 39(4):24–27, 1996.

[16]. A. Karwath and K. Kersting. Relational sequence alignments and logos. pages 290–304, 2007.

[17]. W. Winiwarter. JETCAT – Japanese-English Translation using Corpus-based Acquisition of Transfer rules. JCP, 2(9):27–36, 2007.

[18]. T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops), pages 63–69, 2002.

AUTHORS

S. H. Patil is Professor and Head with Comp. Deptt.,Bharati Vidyapeeth Deemed University College of Engineering, Pune-India. He did his B.E in 1989 from W.I.T. Solapur, Shivaji University, M.E. from University of Pune in 1992 and Ph.D. in Bharti Vidyapeeth Deemed University in 2009 in the area of Operating System. He has completed four research projects under AICTE Research Promotion Schemes.

Prakash Devale is Professor and Head with IT. Deptt., Bharati Vidyapeeth Deemed University College of Engineering, Pune-India. He did his B.E in Computer Science, M.E. from Bharati Vidyapeeth Deemed University College of Engineering, Pune in 2002 and Ph.D. Pursuing in Bharti Vidyapeeth Deemed University in the area of Machine Translation. His Publication in International Journals : 33, Publication in International Conference : 09, Publication in National Conference : 17. He is lifetime member of ISTE.

Sandeep Warhade is a student of M.Tech. IT. Deptt. Bharati Vidyapeeth Deemed University College of Engineering, Pune-India. He did his B.E in Computer Science in 2000. He is doing project in Statistical Machine Translation. He has published two International Journals.