Statistical Machine Translation using Joshua:
An approach to build "enTel" system

[+]Anitha Nalluri,*Vijayanand Kommaluri
+*Advisory System Analyst*, IBM India, Bangalore, India.
**Assistant Professor*, Dept of Computer Science, Pondicherry University, India
Email:analluri@in.ibm.com

1.0 Abstract

This paper addresses an approach to build "enTel" System – An English to Telugu Machine Translation (MT) System using Statistical Machine Translation (SMT) techniques and Johns Hopkins University Open Source Architecture (JOSHUA). It provides a heuristic approach - To train a probabilistic alignment model and use its predictions to align words and ensure the well form of the target language sentences - The tuning of weights of model to balance the contribution of each of the component parts to find the optimal weights among different models – Evaluation of the quality of machine translation with the Bilingual Evaluation Understudy (BLEU) that compares a system's output against reference human translations.

2.0 Introduction

Machine translation (MT), also known as "automatic translation" or "mechanical translation," is the name for computerized methods that automate all or part of the process of translating from one human language to another. Languages are challenging, because natural languages are highly complex, many words have various meanings and different possible translations, sentences might have various readings, and the relationships between linguistic entities are often vague. The major issues in MT involve ambiguity, structural differences between languages, and multiword units such as collocations and idioms. If sentences and words only had one interpretable meaning, the problem of interlingual translation would be much easier. However, languages can present ambiguity on several levels. If a word can have more than one meaning, it is classified as lexically ambiguous. An approach to solving this problem is statistical analysis.

3.0 Statistical Machine Translation

Statistical Machine Translation (SMT) is founded on the theory that every source language segment has any number of possible translations, and the most appropriate is the translation that is assigned the highest probability by the system. It requires a bilingual corpus for each language pair, a monolingual corpus for each target language, a language modeler and a decoder. A language model analyses the monolingual TL corpus in order to 'learn' a sense of grammaticality (e.g. word order), based on n-gram statistics (usually trigrams), and then calculates the probabilities of word x following word y etc. in the TL. The probabilities are calculated during the preparation stage and stored. When presented with a new translation, the SL segments are segmented into smaller phrases. They are matched with source language equivalents in the corpus and their translations harvested by the decoder. As the search space is theoretically infinite, the decoder uses a heuristic search algorithm to harvest and select appropriate translations. The translation problem can be describes as modeling the probability distribution Pr(E|T) Where E is the string in Source language and T is the string in Target Language.

$$\mathrm{Pr}(\mathrm{E}|\mathrm{T}) = \frac{\mathrm{Pr}(\mathrm{T}|\mathrm{E})\mathrm{Pr}(\mathrm{E})}{\mathrm{Pr}(\mathrm{T})}$$

Where, $\mathrm{Pr}(\mathrm{E})$ is called Language Model (LM) and $\mathrm{Pr}(\mathrm{T}|\mathrm{E})$ is called Translation Model (TM).

The use of statistical techniques in machine translation has led to dramatic improvements in the quality of research systems in recent years. The statistical machine translation is rapidly progressing, and the quality of systems is getting better and better. An important factor in these improvements is definitely the availability of large amounts of data for training statistical models. Yet the modeling, training, and search methods have also improved since the field of statistical machine translation was pioneered by IBM in the late 1980s and early 1990s.

3.1 N-GRAM Modeling

An n-gram is a subsequence of n items from a given sequence. The items in question can be phonemes, syllables, letters, words or base pairs according to the application. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram"; and size 4 or more is simply called an "n-gram". Some language models built from n-grams are "(n − 1)-order Markov models". An n-gram model is a type of probabilistic model for predicting the next item in such a sequence. N-gram models are used in various areas of statistical natural language processing and genetic sequence analysis. The n-gram model, a special type of a Markov model, predicts the occurrence of the ith word vi with the formula:

$P(v_i) = [\ c(v_i - (n\text{-}1) \ldots v_i)\ ] / [c(v_i - (n\text{-}1) \ldots v_{i-1})]$

In this formula, c(x) is the number of occurrences of event x. The most significant results in SBMT have been achieved using n-gram modeling and the most common approach is the trigram model, where n = 3.

3.2 SRILM

SRILM is a collection of C++ libraries, executable programs, and helper scripts designed to allow both production of and experimentation with statistical language models for speech recognition and other applications. The toolkit supports creation and evaluation of a variety of language model types based on N-gram statistics, as well as several related tasks, such as statistical tagging and manipulation of N-best lists and word lattices.

3.3 GIZA++

GIZA++ is the Statistical Machine Translation toolkit which was developed by Statistical Machine Translation Team during the summer workshop in 1999 at the Center for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU). It is an extension of the program GIZA (part of the SMT toolkit EGYPT). GIZA ++ is used to train the IBM models 1-5 and HMM Word Alignment model and various smoothing techniques for fertility, distortion/alignment parameters. The training of the fertility models is significantly more efficient.

3.4 Bilingual Evaluation Understudy

The primary programming task for a Bilingual Evaluation Understudy (BLEU) is to compare n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches are position independent. The more the matches, the better the candidate translation is. BLEU's strength is that it correlates highly with human judgments by averaging out individual sentence judgment errors over a test corpus rather than attempting to divine the exact human judgment for every sentence: quantity leads to quality[1]. Thus the BLEU method is used for evaluation of quality of machine translation systems.

4.0 Overview of JOSHUA Architecture

Joshua is an open-source toolkit for parsing-based machine translation that is written in Java. JOSHUA decoder assumes a probabilistic synchronous context-free grammar (SCFG) [2]. During decoding, each time a rule is called to construct a new constituent, a number of feature functions are called in order to give a cost for that constituent.

4.1 Translation Grammars

There are a series of classes which define how grammars are created and used. Initially a Grammar Factory to be constructed to handle the intricacies of parsing grammar files in order to produce a Grammar. This separation is used to decouple the file format from the in-memory representation with the same data structure but different file parsers. The Grammar mostly serves as a wrapper around TrieGrammar in order to give a holistic object representing the entire grammar, though it also gives a place to store global state which would be inappropriate to store in each TrieGrammar object. The TrieGrammar implements a trie-like interface for representing dotted rules for use in parsing charts. This abstract trie can also be viewed as an automaton. Each state of the automaton is represented by a TrieGrammar object [3].

RuleCollection is a collection of individual Rule objects. If these states of TrieGrammar objects are "final" then there is a RuleCollection which could be applied at the current position in parsing. This RuleCollection gives the candidate set of rules which could be applied for the next step of the chart-parsing algorithm. Each of these rules is passed to the PhraseModelFF feature function which will produce the cost for applying that rule [3].

4.2 Convolution

Sometimes for simple implementations this detailed separation of GrammarFactory, Grammar, TrieGrammar, and RuleCollection may seem like overkill. An important thing to keep in mind is that since these are all interfaces, a given implementation can have a smaller number of classes which implement more than one interface [3].

4.3 Language Models

Similarly there are a number of classes that play into language modeling. The NGramLanguageModel interface defines what it means to be a language model. An object of that type is given to the LanguageModelFF feature function which handles all the dynamic programming and N-gram state maintenance [3].

4.4 Minimum Error Rate Training

To balance the contribution of each of the component parts (language model probability, translation model probabilities, lexical translation probability, etc) of the model, the weights should tune to run Minimum Error Rate Training (MERT) for finding the optimal weights among different models [3].

4.5 Evaluation of Translation Quality

The quality of machine translation is commonly measured using the BLEU metric, which automatically compares a system's output against reference human translations. The BLUE metric can be computed using built-in function of "JoshuaEval" [3]. The translation quality can be further improved by varying the size and weights of training data.

5.0 Machine Translation Systems – Telugu Language – Scenario

Telugu is classified as a Dravidian language with heavy Indo-Aryan influence spoken in the Indian state of Andhra Pradesh. Telugu has the third largest number of native speakers in India (74 million according to the 2001 census) and is 15th in the Ethnologue list of most-spoken languages worldwide.

Sampark – Machine Translation among Indian Languages developed by the consortium of 11 Indian institutions led by International Institute of Information Technology-Hyderabad (IIIT-H) is slated for national launch [4]. It can also translate entire webpage with pictures and graphics intact. Anusaaraka - A machine Translation system has been built from Telugu, Kannada, Bengali, Punjabi and Marathi to Hindi [5]. It is domain free but the system has been applied mainly for translating children's stories. Anubharti - A machine-aided-translation is a hybridized example-based machine translation approach that is a combination of example-based, corpus-based approaches and some elementary grammatical analysis. The example-based approaches follow human-learning process for storing knowledge from past experiences to use it in future [6]. AnuBharti II - the traditional EBMT approach has been modified to reduce the requirement of a large example-base. This is done primarily by generalizing the constituents and replacing them with abstracted form from the raw examples. Matching of the input sentence with abstracted examples is done based on the syntactic category and semantic tags of the source language structure[7].

6.0 Development of "enTel" System

An "enTel" system using Joshua is developed and piloted to find the feasibility and effectiveness of statistical machine translation system between English- Telugu languages. A parallel corpus of south Asian languages called Enabling Minority Language Engineering (EMILLE) for Telugu Language developed by the Central Institute for Indian Languages, Mysore, India and "English to Telugu Dictionary" developed by Charles Philip Brown is considered for training of datasets. The language model is trained using SRILM and GIZA++ tools. The size and weights of training data are tuned to achieve the better quality of machine translation system. The quality of the machine translation system is assessed using BLUE metric.

7.0 Conclusion

The piloted “enTel” System is observed to be an efficient and feasible solution of open MT system for English to Telugu. The “enTel” system requires more enormous amounts of parallel text in the source and target text to achieve high quality translation. SMT gives better results as more and more training data is available. The future work of enTel system is proposed to develop the user interfaces that can retrieve the translated text from source language to targeted language with an ease of clicking a mouse.

8.0 References

[1] *Kishore Papineni etal., (2002),* “BLEU: a Method for Automatic Evaluation of Machine Translation”, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.*

[2] *Zhifei Li etal., (2009)*, “Joshua: An Open Source Toolkit for Parsing-based Machine Translation”, *Proceedings of the Fourth Workshop on Statistical Machine Translation , pages 135–139, Athens, Greece, 30 March – 31 March 2009.*

[3] http://www.clsp.jhu.edu/wiki2/Joshua_architecture, *Site last visited 2nd October 2010.*

[4] http://syedakbarindia.blogspot.com/2010/08/iiit-hyderabad-develops-machine.html, *Site last visited 2nd October 2010.*

[5] *Rajeev Sangal etal., (1997)* “ANUSAARAKA: Machine Translation in Stages” , *Appeared in Vivek - A Quarterly in Artificial Intelligence, Vol.10, No.3 (July 1997), NCST, Mumbai, pp.22-25.*

[6] *Renu Jain etal., (2001),* “ANUBHARTI: Using Hybrid Example-Based Approach for Machine Translation”, Proc. Symposium on Translation Support Systems (STRANS2001), February 15-17, 2001, Kanpur, India.

[7] *R.M.K. Sinha (2004)* “An Engineering Perspective of Machine Translation: AnglaBharti-II and AnuBharti-II Architectures”, *Proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS- 2004), November 17-19, 2004, Tata Mc Graw Hill, New Delhi.*