N-Gram Approach for Syntatic (Syntax) Transformation in Cross Language MT

Kathiravan. P[1], Makila. S[2], Prasanna. H[3], Vimala. P[4]
[1, 2, 3, 4] Department of Computer Science
[1, 2, 3, 4] Thiru.Vi.Ka Govt Arts College, Tiruvarur, Tamilnadu, India.

Abstract- *Tamil and English are two entirely different languages, belonging to two different families. Their vocabularies, pronunciation, grammatical structures, syntax(order of words), inflectional patterns(word endings) are absolutely different. If Tamil to Malayalam, Telugu or Kannada translation process is easy, because these languages belong to one family and have many common characteristics. But Tamil-English :No! In Tamil, usually the verb comes at the end of a sentence. AVAN KOVILUKKU POGERAN [s,o,v] pattern But in English, verb comes next to subject. HE GOES TO TEMPLE [s,v,o] pattern. This is a major difference between these languages. In this paper, I concentrate in above criteria by using N-gram (Number of Grammar) implementation.*

Keywords- NLP, machine translation, transliteration , n-gram, syntax transfer.

I. INTRODUCTION

This Section formally defines the goal of this whole paper. Syntax-based translation is based on the idea of translating syntactic units, rather than single words or strings of words (as in phrase-based MT), i.e. (partial) parse trees of sentences/utterances. The idea of syntax-based translation is quite old in MT, though its statistical counterpart did not take off until the advent of strong stochastic parsers in the 1990s. Examples of this approach include DOP-based MT and, more recently, synchronous context-free grammars.

II.SYNTAX

OUT OF VOCABULARY (OOV) WORDS

SMT systems store different word forms as separate symbols without any relation to each other and word forms or phrases that were not in the training data cannot be translated. This might be because of the lack of training data, changes in the human domain where the system is used, or differences in morphology.

Statistical translation systems works in two stages viz. training and translation. In training it ―learns how various languages work. Before translation, the system must be trained. Training is done by feeding the system with source language documents and their high-quality human translations in target language. With its resources, the system tries to guess at documents meanings. Then an application compares the guesses to the human translations and returns the results to improve the system‘s performance. The whole process is carried out by dividing sentences into N-grams. While training, statistical systems track common N-grams, translations most frequently used are learnt and those meanings when finding the phrases in the future are applied. They also statistically analyze the position of N-grams in relation to one another within sentences, as well as words grammatical forms, to determine correct syntax. After their training, the systems are used to process actual phrases and produce the translation from what ever it has learnt in training phase.

III. MOTIVATION AND GOALS

A reasonable domain specific English-Tamil MT can find its immediate applications in government and education sector.

The main goal of this work is to study about English-Tamil machine translation system using rule-based and corpus-based approaches. For rule based approach, considering the structural difference between English and Tamil, syntax transfer based methodology is adopted for translation.

IV. COMPARATIVE STUDY OF ENGLISH AND TAMIL

4.1.1 STRUCTURE OF ENGLISH AND TAMIL LANGUAGES

English belongs to the Indo-European family of languages. Geographically English is the most widespread language on the earth and is second only to Mandarin Chinese with regard to the number of speakers. Modern English is analytic. It has pre-positions. Tense and time in English are indicated by auxiliaries that are always placed before the main verbs. In interrogative sentences auxiliaries are shifted to the front position. Adjectives and noun qualifiers always precede the nouns they qualify. It is the only European language which employs uninflected adjectives. Syntactically English is an SVO language. The word order is rather rigid and fixed in English. In complex sentences, the subordinate clause follows the main clause.

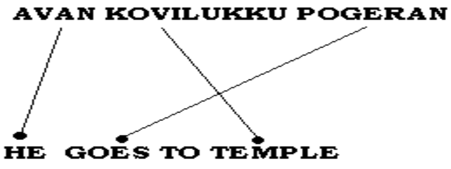


Fig 4.1 Structure in Tamil and English

Table 4.1.1 SVO Pattern in English

SUBJECT	VERB	OBJECT
He	Go	Temple

Table 4.1.2 SOV Pattern in Tamil

SUBJECT	OBJECT	VERB
Avan	Kovilukku	Pogeraan

4.1.1.1 AMBIGUITY

Words and phrases in one language often map to multiple words in another language.

For example, in the sentence,
I went to the bank,
it is not clear whether the “mound of sand” (*Karai in Tamil*) sense or the “financial

Also, each language has its own idiomatic usages which are difficult to identify from a sentence. For example,
Do not beat about the bush, come to the point.

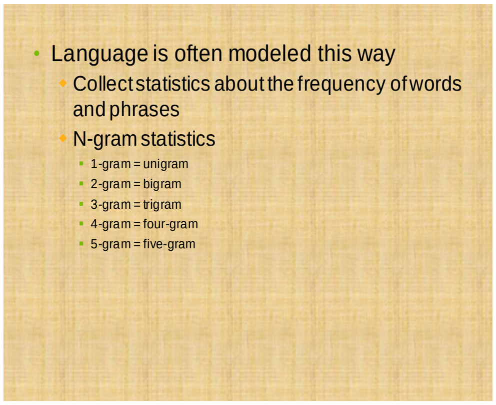
Yet another kind of ambiguity that is possible is structural ambiguity:

He goes to temple.

This can be translated in Tamil as either of the following two sentences.
Avan Pogeran kovilukku
Avan kovilukku pogeran

V. LANGUAGE MODELING USING N-GRAMS

For computers, the easiest way to break a string down into components is to consider substrings. An n-word substring is called an n-gram. If n=2, we say bigram. If n=3, we say trigram. If n=1, nerds say unigram, and normal people say word.



- Language is often modeled this way
  - Collect statistics about the frequency of words and phrases
  - N-gram statistics
    - 1-gram = unigram
    - 2-gram = bigram
    - 3-gram = trigram
    - 4-gram = four-gram
    - 5-gram = five-gram

Fig 5.1 example to N-gram

If a string has a lot of reasonable n-grams, then maybe it is a reasonable string. Not necessarily, but maybe.

Let b(y | x) be the probability that word y follows word x. We can estimate this probability from online text. We simply divide the number of times we see the phrase “xy” by the number of times we see the word “x”. That's called a conditional bigram probability. Each distinct b(y | x) is called a parameter.

A commonly used n-gram estimator looks like this:

b(y | x) = number-of-occurrences(“xy”) / number-of-occurrences(“x”)

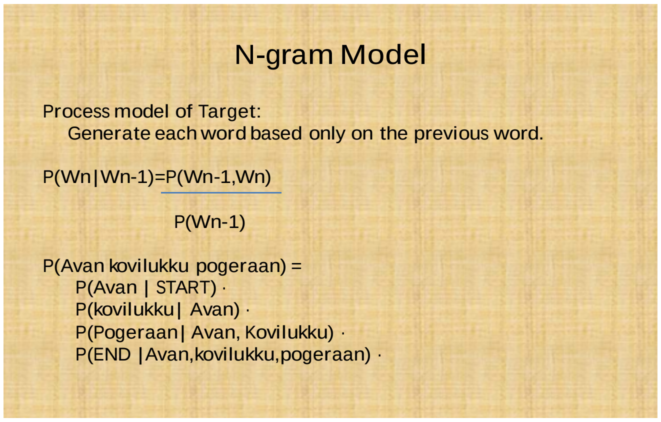
P(I like snakes that are not poisonous) ~
b(I | start-of-sentence) *
b(like | I) *
b(snakes | like) *
...
b(poisonous | not) *
b(end-of-sentence | poisonous)

In other words, what's the chance that you'll start a sentence with the word “I”? If you did say “I”, what's the chance that you would say the word “like” immediately after? And if you did say “like”, is “snakes” a reasonable next word? And so on.

Actually, this is another case of a generative model**.** This model says that people keep spitting out words one after another, but they can't remember anything except the last word they said. That's a crazy model. It's called a bigram language model. If we're nice, we might allow for the possibility that people remember the last two words they said. That's called a trigram language model:

b(z | x y) = number-of-occurrences("xyz") / number-of-occurrences("xy")

P(I like snakes that are not poisonous) ~
  b(I | start-of-sentence start-of-sentence) *
  b(like | start-of-sentence I) *
  b(snakes | I like) *
  ...
  b(poisonous | are not) *
  b(end-of-sentence | not poisonous) *
  b(poisonous | end-of-sentence end-of-sentence)



N-gram Model

Process model of Target:
  Generate each word based only on the previous word.

$$P(Wn|Wn-1)=\frac{P(Wn-1,Wn)}{P(Wn-1)}$$

P(Avan kovilukku pogeraan) =
  P(Avan | START) ·
  P(kovilukku| Avan) ·
  P(Pogeraan| Avan, Kovilukku) ·
  P(END |Avan,kovilukku,pogeraan) ·

Fig 5.2 words broken by n-gram model

The following example comes from the above n-gram approach.

Fig 5.2 the broke words make several sentences and SMT choose the probable sentence using mathematical formula.

n-gram approach gives several outputs and the SMT choose the correct output using the probability for example Bayes rule give the accurate output which one is probably mach to source to target language.

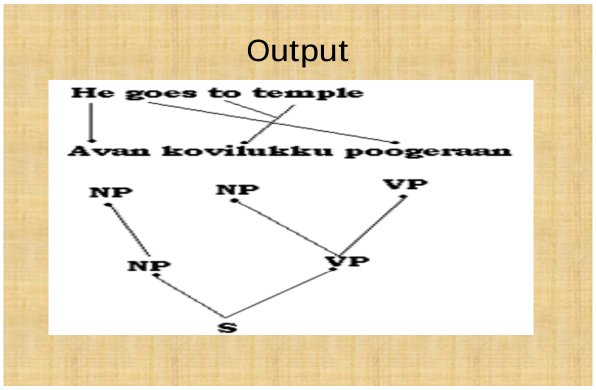


Fig 5.3 output model

The output or our translation (syntax transfer) probably doing like above example.

He goes to temple == Avan kovilukku pogeraan.
English==Tamil

VI. CONCLUSION

In general the technic of MT needs large amounts of linguistic knowledge to be encoded as rules. Particularly the syntactic is very important during the cross language translation in this problem will satisfied by using n-gram approach according to the language model.

REFERENCES

[1] W.John Hutchins and Harold L.Somers .,"An Introduction to Machine Translation ", Centre for computational linguistics .,UK

[2] F.Brown, Stephen A.Della Pietra ,"The Mathematics of Statistical Machine Translatin: Parameter Estimation" IBM T.J Watson Research Center,

[3] Basir Ahmed, Sung-Hyuk Cha, and Charles Tappert. ,"Language Identification from text using N-gram based cumulative frequency addition "

[4] Peter F.Brown, John Cocke, Stephen A.Della Pietra, Vincent J.Della Pietra ,"Statistical approach to machine translation" IBM .

[5] Peter Nather., "N-gram based Text Categorization", Comenius University, 2005

[6] Renu Jain, R.M.K Sinha and Ajai Jain.2001. ANUBHARTI: Using Hybrid Example-Based Approach for Machine Translation in Proc. STRANS2001, February 15-17, Kanpur, India.

[7] Fazly and S. Stevenson. 2006 Automatically constructing a lexicon of verb phrase idiomatic combinations. In proceedings of EACL-2006. pp. 156-178.

[8] R.M.K. Sinha. 1984. Computer processing of Indian Languages and Scripts-Potentialities and problems, Jour. Of Inst.Electron & Telecom.(India).

[9] Gobinda G.Chowdhury, Natural Language Processing, University of strathclyde, Glasgow G1 1XH, UK.