SEMESTER PROJECT

STATISTICAL MACHINE TRANSLATION

May 22, 2017

Guided By : Manish Shrivastava
201506586 Devi Sravani Simhadri
201505613 Rachita Sowle
201505614 Akanksha Singh
IIIT Hyderabad

Contents

INTRODUCTION

Statistical machine translation (SMT) is a machine translation paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora.

Statistical Machine Translation has a huge cost involved as the order of corresponding words are different in source language sentence and target language sentence. To compensate this , we reorder the source sentence according to the target sentence order , i.e. for example as English follows SVO , while Hindi follows SOV , so we will reorder English sentence as SOV and then pass it on to moses translator. This reordered sentence will also increase the accuracy of translation apart from reducing the cost of translation.

In this project we have used Giza alignment tool to get the alignments of words and Constituency parser of Stanford , to get the chunks(phrases) of English. Using outputs of these two , we have derived two algorithms , which will predict the reordering rules for each language pairs.

WORD ALIGNMENT : GIZA

Bitext word alignment or simply word alignment is the natural language processing task of identifying translation relationships among the words (or more rarely multiword units) in a bitext, resulting in a bipartite graph between the two sides of the bitext, with an arc between two words if and only if they are translations of one another. Word alignment is typically done after sentence alignment has already identified pairs of sentences that are translations of one another.

Bitext word alignment is an important supporting task for most methods of statistical machine translation; the parameters of statistical machine translation models are typically estimated by observing word-aligned bitexts, and conversely automatic word alignment is typically done by choosing that alignment which best fits a statistical machine translation model. Circular application of these two ideas results in an instance of the expectation-maximization algorithm.

This approach to training is an instance of unsupervised learning, in that the system is not given examples of the kind of output desired, but is trying to find values for the unobserved model and alignments which best explain the observed bitext. Recent work has begun to explore supervised methods which rely on presenting the system with a (usually small) number of manually aligned sentences.[3] In addition to the benefit of the additional information provided by supervision, these models are typically also able to more easily take advantage of combining many features of the data, such as context, syntactic structure, part-of-speech, or translation lexicon information, which are difficult to integrate into the generative statistical models traditionally used.

Besides the training of machine translation systems, other applications of word alignment include translation lexicon induction, word sense discovery, word sense disambiguation and the cross-lingual projection of linguistic information.

CHUNKING

Chunking in NLP is Changing a perception by moving a âĂIJchunkâĂİ, or a group of bits of information, in the direction of a Deductive or Inductive conclusion through the use of language.

Chunking up or down allows the speaker to use certain language patterns, to utilize the natural internal process through language, to reach for higher meanings or search for more specific bits/portions of missing information.

When we "Chunk Up" the language gets more abstract and there are more chances for agreement, and when we "Chunk Down" we tend to be looking for the specific details that may have been missing in the chunk up.

As an example if you ask the question "for what purpose cars?" you may get the answer "transport", which is a higher chunk and more toward abstract.

If you asked "what specifically about a car"? you will start to get smaller pieces of information about a car.

Lateral thinking will be the process of chunking up and then looking for other examples: For example "for what intentions cars?", "transportation", "what are other examples of transportation?" "Buses!"

Proposed Algorithms

Algorithm 1 Marking Nodes Parent

```
procedure
    $p1 \leftarrow$ Original ordered source sentence
    $p2 \leftarrow$ Original ordered target sentence
    $q1 \leftarrow$ Giza(p1,p2)
    // where q1 is reodered source sentence according to Giza alignments
Numbering:
    $l1 \leftarrow$ Each word of p1 is assigned it's position in sentence as it's number.
    $l2 \leftarrow$ Position of q1's word in p1
For each Non-Leaf node in Constituency tree of p1 :
    $x \leftarrow$ Maximum order number among the right child of p1 sentence
    $y \leftarrow$ Maximum order number among the right child of q1 sentence
    $r \leftarrow$ Maximum order number among the left child of p1 sentence
    $t \leftarrow$ Maximum order number among the left child of q1 sentence
    $value \leftarrow$ (x-y) + (r-t)
    Mark the node with the value and store the node and it's child with the value
For each new Non-leaf node and it's children in data test set :
    If this structure is marked negative in data set , then their is a swap.
    If this structure is marked positive in data set , then their is a no swap.
```

Algorithm 2 Naive based

```
procedure
    p1 $\leftarrow$ Original ordered source sentence
    p2 $\leftarrow$ Original ordered target sentence
    q1 $\leftarrow$ Giza(p1,p2)
    // where q1 is reodered source sentence according to Giza alignments
Numbering:
    l1 $\leftarrow$ Each word of p1 is assigned it's position in sentence as it's number.
    l2 $\leftarrow$ Position of q1's word in p1
Cases: For each word A of p1 and B of q1
    Swap : The numbers of two words have been exchanged
    A goes before B : The order of numbers of A and B have been changed , with lower order number nov
    A goes after B : The order of numbers of A and B have been changed , with higher order number now
    No swap : Their is no change in the number of A and B
Conditions :
    C $\leftarrow$ A and B's lowest common ancestor in constituency tree of p1
    S $\leftarrow$ C's sibling in constituency tree of p1
    GP $\leftarrow$ C's parent in constituency tree of p1
```