Statistical Machine Translation

Hicham El-Zein
Jian Li

University of Waterloo

May 20, 2015

Overview

Introduction

- **Machine Translation** is the problem of automatically translating from one language (source language) to another language (target language).
- It is one of the oldest problems in Artificial Intelligence and Computer Science.
- It is a problem that has huge impact and implications.

Challenges in Machine Translation

- **Lexical Ambiguity:** a word can have distinct meanings. For example:
  - book the flight vs read the book
  - the box was in the pen vs the pen was on the table
- **Different word orders.**
  - English word order: subject - verb - object
  - Japanese word order: subject - object - verb
  - English: The dog saw the cat.
  - Japanese: The dog the cat saw.

Challenges in Machine Translation

- **Syntactic Structure is not Preserved Across Translations**
  - English: The bottle floated into the cave.
  - Spanish: La bottela entro a la cuerva flotando. (The bottle entered the cave floating.)
- Floated was a verb in English got translated to an adverb(flotando) in Spanish.
- Into a proposition was translated to the main verb entered(entro) in Spanish.

Challenges in Machine Translation

- **Syntactic Ambiguity** causes problems.
- For example the sentence: ‘Call me a cab.’ has two different meanings.
- **Pronoun Resolution**.
  - ‘The computer outputs the data, it is fast.’
  - ‘The computer outputs the data, it is stored in asci.’
- It can refer to the computer or to the data.
- We will have different translations for each possibility.

Classical Machine Translation

- We will give a high level description of the classical machine translation systems.
- These systems are rule based systems.

Direct Machine Translation

Direct Machine Translation:

- Translation is done word by word.
- Very little analysis of source text.
- Relies on a large bilingual dictionary. For each word in the source language, the dictionary specifies a set of rules for translating that word.
- After the words are translated, simple reordering rules are applied (e.g., move adjectives after nouns when translating from English to French)

Direct Machine Translation

The lack of any analysis of the source language in Direct Machine Translation causes some problems, for example:

- It is difficult or impossible to capture long range reordering.
- Words are translated without any disambiguation of their syntactic role.

Transfer-Based Approaches

Transfer-Based Approaches: Done in three phases.

- Analysis: Analyze the source language sentence; for example, build a syntactic analysis of the source language sentence.
- Transfer: Convert the source-language parse tree to a target-language parse tree.
- Generation: Convert the target-language parse tree to an output sentence.

Transfer-Based Approaches

- The parse trees involved can vary from shallow analyses to much deeper analyses.
- The transfer rules might look quite similar to the rules for direct translation systems. But they can now operate on syntactic structures.
- It is easier with these approaches to handle long-distance reordering.

Interlingua-Based Translation

Interlingua-Based Translation: Done in two phases.

- Analysis: Analyze the source language sentence into a (language-independent) representation of its meaning.
- Generation: Convert the meaning representation into an output sentence.

Interlingua-Based Translation

- Advantage: If we want to build a translation system that translates between k languages, we need to develop k analysis and generation systems. With a transfer based system, we'd need to develop $O(k^2)$ sets of translation rules.
- Disadvantage: What would a language-independent representation look like?

Interlingua-Based Translation

- How to represent different concepts in a unified language?
- Different languages break down concepts in quite different ways:
  - German has two words for wall: one for an internal wall, one for a wall that is outside.
  - Japanese has two words for brother: one for an elder brother, one for a younger brother.
  - Spanish has two words for leg: one for a human's leg, and the other for an animal's leg, or the leg of a table.
- A unified language may be the intersection of all languages, but that doesn't seem very satisfactory.

Introduction to Statistical Machine Translation

- Motivation: parallel corpora are available in several language pairs
- Basic idea: use a parallel corpus as a training set of translation examples
- Examples:
  - IBM work on French-English translation using the Canadian Hansards (1.7 million sentences of 30 words or less in length)
  - Canadian parliament, English-French
  - Europarl
- Idea goes back to Warren Weaver (1949): suggested applying statistical and cryptanalytic techniques to translation

Introduction to Statistical Machine Translation

. . . one naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say: This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.
(Warren Weaver, 1949, in a letter to Norbert Wiener)

The Noisy Channel Model

- The noisy channel model is a framework used in spell checkers, question answering, speech recognition, and machine translation.
- It is mainly used in spell checkers, but it is still a simple machine translation model.
- Goal: translation system from source language(e.g., French) to target language(e.g., English), $f \rightarrow e$

The Noisy Channel Model

- Have a model $p(e|f)$ which estimates conditional probability of any English sentence e given the French sentence f. Use the training corpus to set the parameters.
- A Noisy Channel Model
  - $p(e)$, the language model
  - $p(f|e)$, the translation model
- Bayes' rule

$$p(e|f) = \frac{p(e,f)}{p(f)} = \frac{p(e)p(f|e)}{p(f)}$$

and

$$\arg\max_e p(e|f) = \arg\max_e p(e)p(f|e)$$

More about the Noisy Channel Model

- The **language model** $p(e)$ could be a trigram model, estimated from any data(parallel corpus not needed to estimate the parameters)
- The **translation model** $p(f|e)$ is trained from a parallel corpus of French/English pairs.
- Note:
  - The translation model is backwards.
  - The language model can make up for deficiencies of the translation model.
  - Challenge: how to build $p(f|e)$
  - Challenge: finding $\arg\max_e p(e)p(f|e)$

Example from Koehn and Knight tutorial

Translation from Spanish to English, candidate translations based on $p(Spanish|English)$ alone:

Que hambre tengo yo

$\rightarrow$

What hunger have $p(s|e) = 0.000014$

Hungry I am so $p(s|e) = 0.000001$

I am so hungry $p(s|e) = 0.0000015$

Have i that hunger $p(s|e) = 0.000020$

...

Example from Koehn and Knight tutorial

With $p(Spanish|English) \times p(English)$:

Que hambre tengo yo
$\rightarrow$
What hunger have $p(s|e)p(e) = 0.000014 \times 0.000001$
Hungry I am so $p(s|e)p(e) = 0.000001 \times 0.0000014$
I am so hungry $p(s|e)p(e) = 0.0000015 \times 0.0001$
Have i that hunger $p(s|e)p(e) = 0.000020 \times 0.00000098$
...

The IBM Translation Models

- IBM Model 1
- IBM Model 2
- EM Training of Models 1 and 2

The IBM Translation Models

- Key ideas in the IBM translation models
  - alignment variables
  - translation parameters, $t(f_i|e_j)$
  - alignment parameters, $q(j|i, l, m)$
- The EM algorithm: an iterative algorithm for training the q and t parameters
- Once the parameters are trained, we can recover the most likely alignments on our training examples
- Recently, the original IBM models are rarely (if ever) used for translation, but they are used for recovering alignments

IBM Model: Alignment

- How do we model $p(f|e)$?
- Assume English sentence e has l words $e_1 \ldots e_l$,
  French sentence f has m words $f_1 \ldots f_m$.
- An alignment a identifies which English word each French word originated from
- Example:
  - English: the dog barks
  - French: le chien aboie
  - An alignment: $a_1 = 1$, $a_2 = 2$, $a_3 = 3$

IBM Model: Alignment

- How do we model $p(f|e)$?
- Assume English sentence e has l words $e_1 \dots e_l$,
  French sentence f has m words $f_1 \dots f_m$.
- An alignment a identifies which English word each French word originated from
- Formally, an alignment a is $\{a_1, \dots, a_m\}$, where each $a_j \in \{0 \dots l\}$.
- There are $(l+1)^m$ possible alignments.

IBM Model: Alignment - Example

- $l = 6,\ m = 7$
- $e =$ And the program has been implemented
- $f =$ Le programme a ete mis en application
- One possible alignment is $\{2, 3, 4, 5, 6, 6, 6\}$
- Another (bad) alignment is $\{1, 1, 1, 1, 1, 1, 1\}$

Alignments in the IBM Models

- Define models for alignment parameter $p(a|e, m)$ and translation parameter $p(f|a, e, m)$

$$p(f, a|e, m) = p(a|e, m)p(f|a, e, m)$$

- Goal

$$p(f|e, m) = \sum_a p(f, a|e, m) = \sum_a p(a|e, m)p(f|a, e, m)$$

An example alignment

- French:
  le conseil a rendu son avis , et nous devons à présent adopter un nouvel avis sur la base de la première position .
- English:
  the council has stated its position , and now , on the basis of the first position , we again have to give our opinion .
- Alignment:
  - the $\rightarrow$ le
  - council $\rightarrow$ conseil
  - has $\rightarrow$ à
  - stated $\rightarrow$ rendu
  - its $\rightarrow$ son
  - position $\rightarrow$ avis
  - , $\rightarrow$,
  - and $\rightarrow$ et
  - now $\rightarrow$ présent
  - , $\rightarrow$ *NULL*
  - on $\rightarrow$ sur
  - the $\rightarrow$ le
  - basis $\rightarrow$ base
  - of $\rightarrow$ de
  - the $\rightarrow$ la
  - first $\rightarrow$ première
  - position $\rightarrow$ position
  - , $\rightarrow$ *NULL*
  - we $\rightarrow$ nous
  - again $\rightarrow$ *NULL*
  - have $\rightarrow$ devons
  - to $\rightarrow$ a
  - give $\rightarrow$ adopter
  - our $\rightarrow$ nouvel
  - opinion $\rightarrow$ avis

A By-Product: Most Likely Alignments

- Once we have a model $p(f, a|e, m) = p(a|e, m)p(f|a, e, m)$, we can also calculate

$$p(a|f, e, m) = \frac{p(f, a|e, m)}{p(f|e, m)} = \frac{p(f, a|e, m)}{\sum_a p(f, a|e, m)}$$

- For a given f, e pair, we can also compute the most likely alignment,

$$a^* = \arg\max_a p(a|f, e, m)$$

IBM Model 1: Alignments

- In IBM model 1 all alignments a are equally likely:

$$p(a|e, m) = \frac{1}{(l+1)^m}$$

- A major simplifying assumption

IBM Model 1: Translation Probabilities

- Next step: find an estimate for

$$p(f|a, e, m)$$

- In model 1, this is

$$p(f|a, e, m) = \prod_{j=1}^{m} t(f_j|e_{a_j})$$

IBM Model 1: Translation Probabilities - Example

- $l = 6, \; m = 7$
- $e =$ And the program has been implemented
- $f =$ Le programme a ete mis en application
- Alignment $a = \{2, 3, 4, 5, 6, 6, 6\}$

$$\begin{aligned} p(f|a, e, m) = & t(Le|the) \times t(programme|program) \times \\ & t(a|has) \times t(ete|been) \\ & t(mis|implemented) \times t(en|implemented) \\ & t(application|implemented) \end{aligned}$$

IBM Model 1: The Generative Process

To generate a French string f from an English string e:

- Step 1: pick an alignment a with probability $\frac{1}{(l+1)^m}$
- Step 2: Pick the French words with probability

$$p(f|a,e,m) = \prod_{j=1}^{m} t(f_j|e_{a_j})$$

The final result:

$$\begin{aligned} p(f,a|e,m) &= p(a|e,m)p(f|a,e,m) \\ &= p(a|l,m)p(f|a,e,m) = \frac{1}{(l+1)^m}\prod_{j=1}^{m} t(f_j|e_{a_j}) \end{aligned}$$

An Example Lexical Entry

- $p(position|position) = 0.7567$
- $p(situation|position) = 0.0548$
- $p(measure|position) = 0.0282$
- $p(vue|position) = 0.0169$
- $p(point|position) = 0.0125$
- $p(attitude|position) = 0.0109$
- $\dots$

- Only difference: we now introduce **alignment or distortion distortion**
  $q(i|j,l,m)$ = probability that j-th French word is connected to i-th English word, given sentence lengths of e and f are l and m respectively
- Define

$$p(a|e,m) = \prod_{j=1}^{m} q(a_j|j,l,m)$$

  where $a = \{a_1, \ldots, a_m\}$
- Gives

$$p(f,a|e,m) = \prod_{j=1}^{m} q(a_j|j,l,m) t(f_j|e_{a_j})$$

IBM Model 2: Example

- $l = 6, \ m = 7$
- $e =$ And the program has been implemented
- $f =$ Le programme a ete mis en application
- $a = \{2, 3, 4, 5, 6, 6, 6\}$

$$\begin{aligned} p(a|e, 7) = & q(2|1, 6, 7) \times q(3|2, 6, 7) \\ & q(4|3, 6, 7) \times q(5|4, 6, 7) \\ & q(6|5, 6, 7) \times q(6|6, 6, 7) \\ & q(6|7, 6, 7) \end{aligned}$$

IBM Model 2: Example

- $l = 6,\ m = 7$
- $e =$ And the program has been implemented
- $f =$ Le programme a ete mis en application
- Alignment $a = \{2, 3, 4, 5, 6, 6, 6\}$

$$
\begin{aligned}
p(f|a, e, 7) = & t(Le|the) \times t(programme|program) \times \\
& t(a|has) \times t(ete|been) \\
& t(mis|implemented) \times t(en|implemented) \\
& t(application|implemented)
\end{aligned}
$$

IBM Model 2: The Generative Process

To generate a French string f from an English string e:

- Step 1: pick an alignment $a = \{a_1, a_2, \ldots, a_m\}$ with probability

$$\prod_{j=1}^{m} q(a_j|j, l, m)$$

- Step 2: Pick the French words with probability

$$p(f|a, e, m) = \prod_{j=1}^{m} t(f_j|e_{a_j})$$

The final result:

$$p(f, a|e, m) = p(a|e, m)p(f|a, e, m) = \prod_{j=1}^{m} q(a_j|j, l, m)t(f_j|e_{a_j})$$

Recovering Alignments

- If we have distributions q and t, we can easily reover the most like alignment for any sentence pair
- Given a sentence pair $e_1, e_2, \ldots, e_l, f_1, f_2, \ldots, f_m$, define

$$a_j = \arg \max_{a \in \{0 \ldots l\}} q(a|j, l, m) t(f_j | e_{a_j})$$

  for $j \in \{1, \ldots m\}$
- the algorithm for recovering alignments is **beam search**

EM Training - the parameter estimation problem

- Input to the parameter estimation algorithm: $(e^{(k)}, f^{(k)})$ for $k = 1 \ldots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence
- Output: parameters $t(f|e)$ and $q(i|j, l, m)$
- The key challenge: **we do not have alignments on our training examples**

Parameter Estimation if the Alignments are Observed

- Example where alignments are observed in training data
  - $e^{(100)}$ = And the program has been implemented
  - $f^{(100)}$ = Le programme a ete mis en application
  - $a^{(100)} = \{2, 3, 4, 5, 6, 6, 6\}$
- Training data is $(e^{(k)}, f^{(k)}, a^{(k)})$ for $k = 1 \dots n$. Each $e^{(k)}$ is an English sentence, each $f^{(k)}$ is a French sentence, each $a^{(k)}$ is an alignment
- Maximum-likelihood parameter estimates in this case are:

$$t_{ML}(f|e) = \frac{Count(e, f)}{Count(e)}$$

$$q_{ML}(j|i, l, m) = \frac{Count(j|i, l, m)}{Count(i, l, m)}$$

Algorithm

Input

A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $\left|f^{(k)}\right| = \left|a^{(k)}\right| = m_k$

Output

$t_{ML}(f|e) = \frac{c(e,f)}{c(e)}$, $q_{ML}(j|i,l,m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Algorithm

- set all counts $c(\ldots) = 0$
- for $k = 1 \ldots n$
  - for $i = 1 \ldots m_k$, for $j = 0 \ldots l_k$,

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k,i,j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k,i,j)$$

$$c(j|i,l,m) \leftarrow c(j|i,l,m) + \delta(k,i,j)$$

$$c(i,l,m) \leftarrow c(i,l,m) + \delta(k,i,j)$$

where $\delta(k,i,j) = 1$ if $a_i^{(k)} = j$; 0, otherwise.

Parameter Estimation with the EM Algorithm

- The algorithm is quiet similar to algorithm when alignments are observed. The only two differences:
  - The algorithm is **iterative**. We start with some initial(e.g., random) choice for the q and t parameters. At each iteration we compute "counts" based on the training data with our current parameter estimates. We then re-estimate our parameters with these counts, and iterate.
  - Computing $\delta(k, i, j)$ by

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k)t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k)t(f_i^{(k)}|e_j^{(k)})}$$

Algorithm

Input

A training corpus $(f^{(k)}, e^{(k)}, a^{(k)})$ for $k = 1 \ldots n$, where $\left|f^{(k)}\right| = \left|a^{(k)}\right| = m_k$

Example

Initialization Initialize $t(f|e)$ and $q(j|i, l, m)$ parameters(e.g., to random values)

Algorithm

Algorithm

- for $s = 1 \ldots S$
  - set all counts $c(\ldots) = 0$
  - for $k = 1 \ldots n$
    - for $i = 1 \ldots m_k$, for $j = 0 \ldots l_k$,

$$c(e_j^{(k)}, f_i^{(k)}) \leftarrow c(e_j^{(k)}, f_i^{(k)}) + \delta(k, i, j)$$

$$c(e_j^{(k)}) \leftarrow c(e_j^{(k)}) + \delta(k, i, j)$$

$$c(j|i, l, m) \leftarrow c(j|i, l, m) + \delta(k, i, j)$$

$$c(i, l, m) \leftarrow c(i, l, m) + \delta(k, i, j)$$

where

$$\delta(k, i, j) = \frac{q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}{\sum_{j=0}^{l_k} q(j|i, l_k, m_k) t(f_i^{(k)}|e_j^{(k)})}$$

Re-calculate the parameters: $t(f|e) = \frac{c(e,f)}{c(e)}$, $q(j|i, l, m) = \frac{c(j|i,l,m)}{c(i,l,m)}$

Details of the Algorithm

- The log-likelihood function

$$L(t, q) = \sum_{k=1}^{n} \log p(f^{(k)}|e^{(k)}) = \sum_{k=1}^{n} \log \sum_{a} p(f^{(k)}, a|e^{(k)})$$

- The maximum-likelihood estimates are

$$\arg\max_{t,q} L(t, q)$$

- The EM algorithm will converge to a *local maximum* of the log-likelihood function

Phrase-Based Translation Overview

- Learning phrases from alignments
- A phrase-based model
- Decoding in phrase-based models

Learning Phrases from Alignments

- First stage in training a phrase-based model is extraction of a **Phrase-Based Lexicon**.
- A Phrase-Based Lexicon pairs strings in one language with strings in another language:
  - nach Kanada $\leftrightarrow$ in Canada
  - zur Konferenz $\leftrightarrow$ to the conference
  - Morgen $\leftrightarrow$ tomorrow
  - ...
- We need to capture the probability distribution $t(e|s)$ where e is a phrase in the target language and s is a phrase in the source language.

Learning Phrases from Alignments

- For example:
  - English: Mary did not slap the green witch
  - Spanish: Maria no daba una bofetada a la bruja verde
- Some (not all) phrase pairs extracted from this example:
- (Mary $\leftrightarrow$ Maria), (no $\leftrightarrow$ did not), (no daba una bofetada $\leftrightarrow$ did not slap).
- We'll see how to do this using alignments from the IBM models.

Learning Phrases from Alignments

- IBM model 2 defines two distributions:
  - $t(s_i|e_j)$ where s_i is a word in the source language and e_j is a word in the target language.
  - $q(i|j,l,m)$ is the probability that the i^{th} word in the source language aligns to the j^{th} word in the target language.
- A useful by-product: once we've trained the model, for any (f,e) pair, we can calculate:

$$\begin{aligned} a^* &= argmax_a p(a|f,e,m) \\ &= argmax_a \prod_{i=1}^{l} q(a_i|i,l,m)t(s_{a_i}|e_i) \end{aligned}$$

under the model. a^* is the most likely alignment.

Learning Phrases from Alignments

	Maria	no	daba	una	bofetada	a	la	bruja	verde
Mary	x								
did						x			
not		x							
slap			x	x	x				
the							x		
green									x
witch								x	

- Every Spanish word is aligned to exactly one English word.
- The alignment is often noisy.
- We need a many to many relation not a one to many relation.

Learning Phrases from Alignments

- Step1: Train IBM model 2 for $p(s|t)$, and come up with the most likely alignment for each (s, t) pair.
- Step2: Train IBM model 2 for $p(t|s)$, and come up with the most likely alignment for each (t, s) pair.
- We now have two alignments, take their intersection as a starting point.

Learning Phrases from Alignments

- Alignment from $p(s|t)$:

	Maria	no	daba	una	bofetada	a	la	bruja	verde
Mary	x								
did						x			
not		x							
slap			x	x	x				
the							x		
green									x
witch								x	

- Alignment from $p(t|s)$:

	Maria	no	daba	una	bofetada	a	la	bruja	verde
Mary	x								
did		x							
not		x							
slap					x				
the							x		
green									x
witch								x	

Learning Phrases from Alignments

Intersection of the two alignments is a very reliable starting point:

	Maria	no	daba	una	bofetada	a	la	bruja	verde
Mary	x								
did									
not		x							
slap					x				
the							x		
green									x
witch								x	

Learning Phrases from Alignments

- Only explore alignment in union of $p(s|t)$ and $p(t|s)$ alignments.
- Add one alignment point at a time.
- Only add alignment points which align a word that currently has no alignment.
- At first, restrict ourselves to alignment points that are neighbours of current alignment points.
- Later, consider other alignment points.

Learning Phrases from Alignments

The final alignment, created by taking the intersection of the two alignments, then adding new points using the growing heuristics:

	Maria	no	daba	una	bofetada	a	la	bruja	verde
Mary	x								
did		x							
not		x							
slap			x	x	x				
the						x	x		
green									x
witch								x	

Note that the alignment is no longer many-to-one: potentially multiple Spanish words can be aligned to a single English word, and vice versa.

Learning Phrases from Alignments

- A phrase-pair consists of a sequence of words, s from the source language, paired with a sequence of words, e from the target language.
- A phrase-pair (s, e) is consistent if:
  - There is at least one word in s aligned to a word in e.
  - There are no words in s aligned to words outside e.
  - There are no words in e aligned to words outside s.
- (Marry did not, Maria no) is consistent, (Marry did, Maria no) is not consistent.
- We extract all pairs from the training example.

Learning Phrases from Alignments

- For any phrase pair (s,e) extracted from the training data, we can calculate:

$$t(e|s) = \frac{Count(s, e)}{Count(s)}$$

Phrase-based Models: Definitions

A **Phrase-Based Model** consists of:

- phrase-based lexicon, consisting of entries (s,e) where each entry has a score $g(s, e) = \lg t(e|s)$.
- A trigram language model.
- A distortion parameter η typically negative.

Definitions

- Given a sentence s in the source language a **Derivation** y is a finite sequence of phrases $p_1, ..., p_L$.
- The length L can be any positive integer value.
- Each phrase $p_i = (s, t, \sigma_1 ... \sigma_m)$ is aligned to the phrase starting from word s and ending in word t in the source sentence.
- A derivation for an input sentence s is valid iff:
  - Each word in s is translated only once.
  - For all $k \in \{1, ..., (L-1)\}, |t(p_k) - s(p_{k+1})| \leq d$ where d is a parameter of the model.
  - Also $|1 - s(p_{k+1})| \leq d$.

Example

- German: wir mussen auch diese kritik ernst nehmen
- y = (1,3, we must also), (7,7, take), (4,5,this criticism), (6,6,seriously)
- y = (1,2, we must), (7,7, take), (3,3, also), (4,5,this criticism), (6,6,seriously)

Scoring Derivations

- The optimal translation under the model for a source-language sentence s will be the valid derivation with the highest score.
- The score of a derivation is defined as:

$$h(y) + \sum_{k=1}^{L} g(p_k) + \sum_{k=0}^{L-1} \eta |t(p_k) + 1 - s(p_{k+1})|$$

- where $h(y)$ is the probability of the sentence y calculated using a tri-gram model.

Example

- wir mussen auch diese kritik ernst nehmen
- y = (1,3, we must also), (7,7, take), (4,5,this criticism), (6,6,seriously)

Decoding Algorithm

- Finding the optimal derivation is an NP-Hard problem.
- We will use a heuristic (Beam Search).
- Beam search is a heuristic search algorithm that explores a graph by expanding the most promising node in a limited set (the node with the highest score).
- Beam search is an optimization of best-first search that reduces its memory requirements.
- Best-first search is a graph search which explores a graph by always expanding the node with the highest score.
- In Beam search, only the nodes with the highest scores are kept as candidates.

Decoding Algorithm - Representing the Search Graph

- A state is a tuple (e_1, e_2, b, r, α) where:
  - e_1, e_2 are english words,
  - b is a bit string of length n,
  - r is an integer specifying the end-point of the last phrase in the state,
  - α is the state score.
- The initial state is: $(*, *, 0^n, 0, 0)$

Decoding Algorithm - Representing the Search Graph

- A state $q = (e_1, e_2, b, r, \alpha)$ is followed by a phrase $p = (s, t, \sigma_1 ... \sigma_m)$ if:
  - p does not intersect b,
  - The distortion limit must not be violated. $(|r + 1 - s(p)| \leq d)$
- The resulting state will be $q' = (e'_1, e'_2, b', r', \alpha')$ where:
  - $e'_1 = \sigma_{m-1}$,
  - $e'_2 = \sigma_m$,
  - $b' = b \cup \{s, ..., t\}$
  - $r' = t$
  - $\alpha' = \alpha + g(p) + \sum_{i=1}^{M} \lg q(\sigma_i | \sigma_{i-2}, \sigma_{i-1}) + \eta |r - s + 1|$

Algorithm

- Now that the states are well defined we use Beam search to find an approximate answer.

The End