# **CSEP 517, Fall 2015, Assignment 4**

<https://github.com/jogonzal-msft/UWNLPAssignment4>

## Introduction

The task at hand is to build three phrase based decoders. Most of the functionality of the program (parsing, identifying and counting phrases, defining a distortion model, getting a language model) is given.

## Implementing the Phrase Decoders

The three decoders that have to be built are the following:

1. Monotonic (without language model)
2. Monotonic (with language model)
3. Non monotonic (with language and distortion model)

Monotonic decoders can only progress in sequential order, meaning they won’t ever consider that phrases can be reordered.

The implementation was done using a 2000 size beam (FastPriorityQueue). The algorithm is the following:

1. Add initial states to the beam (for monotonic, this only adds the phrases that are at the beginning of the sentence, for non-monotonic, it adds all of them)
   1. See DecoderBase. AddInitialStatesToBeam
2. Enter the loop (DecoderBase.DecodeFrenchSentence)
   1. Clean the beam – only process the next 2000 elements. For each of those elements:
   2. Find all “0” sections in the sentence (word “sections” that have not been translated. For each of them:
   3. Find all phrases in them and generate new states for them – put them in the beam
   4. Repeat (a) until the beam is empty, while keeping track of the best last state

## Implementation details

### “Dead end” states pruning

On step 2.b, don’t add states that are dead ends. Dead ends happen when introducing the distortion model, they are state sequences in which non-translated words are so far appart it will be impossible to jump between them, for example: “0011111111100”. See my answer in [the discussion board](https://catalyst.uw.edu/gopost/conversation/arala/942014).

### Distortion and phrase length limit

For both distortions and phrases, use the distortionLimit and maxPhraseLength to limit the iterations and avoid unnecessary calculations.

### Monotonic decoders

Monotonic decoders will only generate one section in 2.b

### Multibeam (partitioned beam)

A partitioned beam was used (meaning an array of beams). The array is of the same length as the French sentence and the beam is selected for a specific state depending on how many French words are translated for that state.

Note: Without the multibeam and “Dead end” states pruning, the non-monotonic (with language and distortion model) approach does not yield the expected results.

## Results

One caveat about these results is that they don’t follow the same beam strategy – for the first Monotonic approaches, a one beam of 2000 is used. For the DIST\_LM approach, a multibeam (see above) is used along with “dead end” pruning (see above).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DecoderIndex | Decoder | BLEU | ModelScore | Processinc time |
| A | Phrase model | 18.982 | -4937.263 | 29.183 |
| B | Phrase and language model | 25.138 | -4074.268 | 1535.357 |
| C | Phrase, language and distortion model | 25.479 | -3775.979 | 32248.763 |

### Further results

When running the DIST\_LM model and using the “one” beam strategy of size 2000 (no multibeam) the maximum score got was around 18. After adding pruning, the score went up to ~24, then after adding multibeam it went to 25.479.

## Analysis

While the DIST\_LM system offers the best score, the implementation offers a very pool performance. I am convinced that further pruning (see “Further possible improvements section below) can be done to improve this processing time and the score in general.

## Common mistakes made by decoder

On the mono\_lm decoder, it is fairly obvious to see that the translation does not make sense in the English language, even though the phrases with the highest likelihood of showing have been selected. In order to add this pseudo-grammatical sense to the sentence, the language model has to be factored in:

MONO\_NOLM: mr president , this that we will , however , have also is to be done at biarritz , that is take a look little more far .

MONO\_LM: mr president , what we will , however , have also done in biarritz , that is to look a little further .

When comparing the MONO\_LM and DIST\_LM, subtleties emerge on sentences where DIST\_LM does better. It is basically on the cases where the reordering yields something that is better for the language model that DIST\_LM does better. This is subtle in this case where “which” Is replaced by other phrases in different positions.

MONO\_LM: we will have to work on the strengthened cooperation , made more easy and effective to achieve this result : the strengthened cooperation which is the way immediate , the more simple to be able to make this step towards the integration , which the need has been affirmed by so many speakers today .

DIST\_LM: we need to work on on closer cooperation, made more easy and effective to achieve this result : enhanced cooperation is the way immediate , the more simple to be able to make this step towards integration , including the need has been affirmed by so many speakers today .

## Further possible improvements

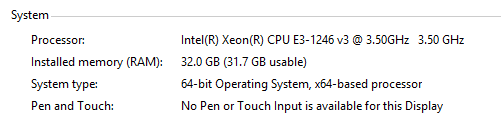
In general I think the decoder can be further improved by doing more pruning in the states (Jurafsky pruning) before adding them to the beam or by adding more space to the Beam (which would impact runtime).

## Code

The class DecoderBase has all the code that adds phrases, it implements a single Beam. DecoderBaseNonMonotonic implements a multibeam. TransitionState is the state class and contains methods to evaluate the state.

## Testing/development notes

Machine specs:



The reported performance numbers were obtained when running with the following configuration on an 8 core 32 gig machine against the test set.

*-cp mt-lib.jar:mt-submit.jar -server -mx2000m -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:+CMSConcurrentMTEnabled*

To run the program, simply comment in/out commands from the main class:

MtDecoderTester.*main*(**new** String[]{  
 **"-path"** ,**".\\bigdata"**,  
 *// "-decoderType", "MONO\_GREEDY"* **"-decoderType"**, **"MONO\_NOLM"** *// "-decoderType", "MONO\_LM"  
 // "-decoderType", "DIST\_LM"*});