Documentation:

Module: 1.py

load (*path*)
Loads parallel corpus which is in json format

Parameters:	**path : str** The file name (string) from where the matrix will be loaded.
Returns:	**data : 2D list** 2D list with Dictionary of both languages as keys

get_tokens (*data*)
Splits sentences into words

Parameters:	**Data: 2D list** 2D list with Dictionary of both languages as keys
Returns:	**Tokens: List** List contains sentence tokenized into words.

iteration **(** *data, tokens, total, prev_prob*)

Loads parallel corpus which is in json format

Parameters:	**Data: 2D list**
	2D list with Dictionary of both languages as keys
	Tokens: List
	Of words by language
	total:
	counts of the destination words, weighted according to their translation probabilities t(e|s)
	Prev_prob:
	Translation probabilities from previous iteration
Returns:	**curr_prob: List**

distance **(table_1,table_2)**

Finds RMS between both the tables.

Parameters:	**table_1**
	table_2
Returns:	**sqrt(res): Integer**

check_convergence (prev_prob, curr_probs, epsilon)
Checks when to stop iterating

Parameters:	**Prev_prob, curr_probs: 2d Dict**
	Translation probability matrices before and after an iteration
	Epsilon: int
	Boundary for convergence.
Returns:	**Bool**
	True if delta<epsilon

train (*data, epsilon*)
Trains the dataset

Parameters:	**Data: 2D list**
	2D list with Dictionary of both languages as keys
	epsilon:int
Returns:	**Curr_probs**
	Translation probability matrices before and after an iteration

Module: 2.py

load (*path*)
Loads parallel corpus which is in json format

Parameters:	**path : str**

	The file name (string) from where the matrix will be loaded.
Returns:	**data : 2D list**
	2D list with Dictionary of both languages as keys

tokData (*data, target='fr'*)

Tokenizes the data and loads it appropriately as 2D list containing Aligned Sent

Parameters:	**data : 2D list**
	2D list with Dictionary of both languages as keys
	target: str
	Language other than english used in corpus loaded.
Returns:	**data : 2D list**
	2D list of tokenized words

main ( )

The main function that carries out all the major steps to produce the results.

No parameters or return values.

Module: 3.py

load (*path*)

Loads parallel corpus which is in json format

Parameters:	**path : str**
	The file name (string) from where the matrix will be loaded.

Returns:	**data : 2D list** 2D list with Dictionary of both languages as keys

phrase_bases_extraction (*filename, foreign*)

Tokenizes the data and loads it appropriately as 2D list containing Aligned Sent

Parameters:	**filename : str** Name of the dataset file **foreign: str** Language other than english used in corpus loaded.

main ( )

The main function that carries out all the major steps to produce the results.

No parameters or return values.