inexactsearch Documentation

Release 0.2

Santhosh Thottingal

September 12, 2013

CONTENTS

This is a Fuzzy string search application. This application illustrates the combined use of Edit distance and Indic Soundex algorithm. By mixing both written like(edit distance) and sounds like(soundex), we achieve an efficient aproximate string searching.

This application is capable of cross language string search too. That means, you can search Hindi words in Malayalam text. If there is any Malayalam word, which is approximate transliteration of hindi word, or sounds alike the hindi words, it will be returned as an approximate match. The "written like" algorithm used here is a bigram average algorithm. The ratio of common bigrams in two strings and average number of bigrams will give a factor which is greater than zero and less than 1. Similarly the soundex algorithm also gives a weight. By selecting words which has comparison weight more than the threshold weight(which 0.6), we get the search results.

CHAPTER

ONE

INEXACTSEACH API

class inexactsearch.core.**InexactSearch**

This class provides methods for fuzzy searching using word distance as well as phonetics.

bigram_average(*str1*, *str2*)

Return approximate string comparator measure (between 0.0 and 1.0) using bigrams. :param str1: string 1 for comparison :str1 type : str :param str2: string 2 for comparison :str2 type : str :returns: int score between 0.0 and 1.0

```
>>> score = bigram_avearage(str1, str2)
0.7
```

Bigrams are two-character sub-strings contained in a string. For example, 'peter' contains the bigrams: pe,et,te,er.

This routine counts the number of common bigrams and divides by the average number of bigrams. The resulting number is returned.

compare(*string1*, *string2*)

Compare strings using soundex if not possible givees biggram avearage.

Parameters

- **str1** (*str.*) – string 1 for comparison.
- **str2** (*str.*) – string 2 for comparison

Returns int score between 0.0 and 1.0

get_info()

Returns info on the module

get_module_name()

returns module name

search(*text*, *key*)

Searches for the key in the given text.

Parameters

- **text** (*str.*) – text in which search has to be done.
- **key** (*str.*) – key which has to be searched

Returns A dictionary with words in the string as keys and the score against the key as the value

inexactsearch.core.**getInstance**()

returns an instance of :class: *InexactSearch* class.

PYTHON MODULE INDEX