LectureHelper

Description:

This program is designed to help Coursera students to see video lectures in text format. It combines 1-minute screenshots of lecture videos with subtitles file to produce textual representation of the lecture.

Here is an example of what the output file looks like:

Lesson1.1

00:00:00.000
This lecture is about Natural Language of Content Analysis. As you see from this picture, this is really the first step to process any text data. Text data are in natural languages. So computers have to understand natural languages to some extent, in order to make use of the data. So that's the topic of this lecture. We're going to cover three things. First, what is natural language processing, which is the main technique for processing natural language to obtain understanding. The second is the state of the art of NLP which stands for natural language processing. Finally we're going to cover the relation between natural language processing and text retrieval. First, what is NLP? Well the best way to explain it is to think about if you see

00:01:02.240
a text in a foreign language that you can understand. Now what do you have to do in order to understand that text? This is basically what computers are facing. So looking at the simple sentence like a dog is chasing a boy on the playground. We don't have any problems understanding this sentence. But imagine what the computer would have to do in order to understand it. Well in general, it would have to do the following. First, it would have to know dog is a noun, chasing's a verb, etc. So this is called lexical analysis, or part-of-speech tagging, and we need to figure out the syntactic categories of those words. So that's the first step. After that, we're going to figure out the structure of the sentence. So for example, here it shows that and the dog would go together to form a noun phrase. And we won't have dog and is to go first. And there are some structures that are not just right.

In addition, in the output directory there is a file called lectures.html. It contains links to all lessons that where processed. Also, for each lecture this file builds summary and a list of key phrases. Here is an example of lectures.html file:

Lectures
Lesson1.1
This lecture is about Natural Language of Content Analysis.
Summary:
This ambiguity can be very hard to disambiguate and you can imagine example where you have to use a lot of knowledge in the context of the sentence or from the background, in order to figure out who actually had the telescope. Of course, put in parentheses but not all, of course there are many queries that are not answered well by the current search engines, and they do require the replantation that would go beyond bag of words replantation. And finally, we also explain why the bag of words replantation remains the dominant replantation used in modern search engines,
Key-phrases:
limited, feedback, So imagine, sum, open topic, such technique, brain, modern search, relation, parsing pretty, major search, subtle, big field, extra knowledge, large knowledge, d1, perfect, machine translation, number, chasing activity, technique, math sense, common sense, communication efficient, program, phrase structure, ambiguity, relevance, exploration, natural language, use something, long run, shallow way, someone, comparison, similar, semantic analysis, general Natural, sense knowledge, verb, course, ambiguous sentence, different meaning, presupposition, world, trouble, speech tagging, comparison such, level, word level, off, work pretty, Language Processing, Text data, entity extraction, try, p, slide, square root, activity, partial understanding, Speech, simple task, user, Again, everyone, Java, telescope, verbal, sequence, word own, attachment ambiguity, hard, Bing, idea, relative, second, computer, result, query, example design, matching, special, Java applet, NLP, language processing, tagging pretty, ambiguous word, bag, semantic matching, simplified view, current search, semantical analysis, reasoning, content, everything, state, computer current, playground, lecture, duplicated, evaluation, certain dataset, complete semantical, biased, Bill, small probability, extend, common example, complete understanding, understanding, difficulty, reason, processing difficult, coffee, extent, imagine, program language, So natural, First, boy, search, restaurant, engine utility, steep, different syntactic, Processing topic, moment, anaphora resolution, word anome, current Natural, place, context, receiver, prepositional phrase, today, complete view, Content Analysis, inference, point, certain meaning,

Installation:

This program is written using Python 2.7

In order to run this program, 2 additional products are required.

1) A program implementing TextEdit algorithm:
   https://github.com/davidadamojr/TextRank

   Just follow the instructions, and it will guide to install all its dependencies.

2) A program for processing video file and converting it to 1-minute slides:
   https://ffmpeg.org/

Preparation:

Before you start this program you need to download Coursera material. You can download lectures from Coursera by hand or using other tools. When you are downloading please maintain the following structure:

```
CourseraRootDirectory
    |_____Lecture1.1
          |_____index.mp4
          |_____subtitle.txt
          |_____subtitles.vtt
    |_____Lecture1.1
          |_____index.mp4
          |_____subtitle.txt
          |_____subtitles.vtt
    .
    .
    .
    |_____Lecture12.1
          |_____index.mp4
          |_____subtitle.txt
          |_____subtitles.vtt
    |_____Lecture12.2
          |_____index.mp4
          |_____subtitle.txt
          |_____subtitles.vtt
```

You can download one or more or all lectures.

Configuration:

This program needs to be configured by adding proper fields to config.py file.

```
courseradir: enter absolute path to CourseraRootDirectory from above.
lecturedir: enter absolute path to output directory.

ffmpegdir:directory where ffmpeg executable is located
```

What happens inside the program:

For each lesson, the program uses TextRank program with subtitle.txt input to create file's summary and a list of keywords.

After this ffmpeg program is called on each lecture's .mp4 file to produce video images (slides) at 1-minute intervals.

Later subtitles.vtt file is combine with slides for this lesson to produce and integrated file.

Output:

The output index file is located at the following place:

OutputDirectory
- images
- lectures
  - lectures.html