Intelligent (Task-Oriented) Conversation Assistant for Course Selection

Progress Report



Information Technology Capstone Project

COMP5703/5707/5708

Group Members

1. Shengyuan Sun (460257820)
2. Rui Chen (470111585)
3. Quan Chen (470199228)
4. Biying Wang (470067400)

Table of Contents

[Table of Contents i](#_Toc521413584)

[1. Progress Status 1](#_Toc521413585)

[2. Roles & Responsibilities 2](#_Toc521413586)

[3. Individual Achievements 2](#_Toc521413587)

[4. Group Collaboration 2](#_Toc521413588)

# Progress Status

|  |  |
| --- | --- |
| **Project Name** | CS17 Intelligence(Task-Oriented) Conversation assistant for course selection |
| **Project Start Date** | 6/ 3/ 2019 |
| **Project Manager** | Biying Wang |

|  |  |
| --- | --- |
| Project Description | The project focuses on using NLP and other technologies to build a dialog system to analyse and answer the questions that students ask in terms of course units on University of Sydney education system |

|  |  |  |
| --- | --- | --- |
| Project Status Report | # | Date: 3/ 5/ 2019 |

|  |  |  |
| --- | --- | --- |
| **Status Item** | **Status up to last week** | **Planned for this week** |
| **Major deliverables** | Intent design  Intent classification | Text preprocessing  Word embedding |
| **Planned delivery date** | 19/ 4 /2019 | 3/5/2019 |
| **Major issues** | Even we have start coding the main part of the project.  One important point is the data and intent type. This week I work with other team member and design the intents | As the intent classification need the input with simple format, so I apply three steps to change the orginal data type. |
| **Major risks** | 1, the first risk is when I design the intent, I think we need design the intent based on the CUSP data structure and other poinion is design the intent with the data base structure. | 1, we discuss the necessary of remove stoping word.  2,the word embedding extend the size of input format, which require more trainng data.  3,the current model testing by batch input, I need modify it and witre a function for following part. |
| **External dependencies** | Slack github | Slack |
| **Estimated effort (hr)** | 20hr | 20hr |
| **Recorded effort (hr)** | 20hr | 20hr |
| **Status (R, Y, G)** | green | green |

# Roles & Responsibilities

Shengyuan Sun

Role: developer

Responsibility:

* Text preprocessing
* modify the intent classification part

Rui Chen

Roles: Analyser and Developer

Responsibility:

* Handbook and CourseMap templates
* Collect data from CUSP website to fill the blank in templates
* Generate more that thirty thousands raw questions with intends and slots
* Dialog Tracker development

Biying Wang

Role: Project Manager; Developer

Responsibility:

* Check weekly deliverables
* Check meeting time, location and meeting topic
* Discuss intent classification and slot filling with team members
* Design Entity list
* Integrate template

Quan Chen

Role: Developer

Responsibility:

* Back-end database system building and data collecting
* Help to design the whole process of system (such as slot and intent)
* Coding for logical part dealing with slot filling and Peewee template code matching.

# Individual Achievements

**1 Tokenization**

Tokenization, in natural language processing, is one of the simple processes to preprocessing the text, the purpose of tokenization is to simplify the text as the input for the following functions of the system. The most common steps of general text tokenization are: converting all letters to lower or upper case， converting numbers into words or removing numbers，removing punctuations, accent marks, and other diacritics，removing white spaces，expanding abbreviations，removing stop words, sparse terms, and particular words and text canonicalization.

In the Cassandra project, the first step is to preprocess the input text, it is important to choose the necessary steps and make sure the result is useful for the intent classification and slot filler part. For Cassandra project, The purpose of preprocessing the text is to omit the length of the sentence, leaving the more important part of the sentence to omit the less useful part of the sentence.

First, we need to convert all letters to lower case. In this step, we first unify the text format so that we can follow the string matching and database operations.

E.g

The user's input is "Hi, I want to know the lecturer address of COMP5426."

The output of this step is "hi, i want to know the lecturer address of comp5426."

Figure 10 convert all letters to lower case

In this sentence, we need to unify "COMP" into a lowercase format so that we can match the background database.

Second, converting numbers words into numbers, for the convenience of back slot matching, so in the preprocessing we need to convert numbers words into numbers.

E.g

"comp five four two six" appears in the user's question,

So in the ability of people to understand, we know that users want to express "comp5426", but it is more convenient to query numbers in the database than words, so you need to turn such digital words into numbers in the process of processing.

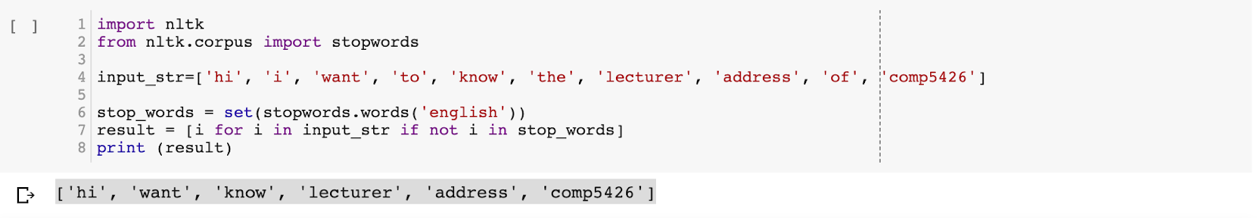
Third, removing stop words, when the user enters a sentence, although these words are necessary grammatically, they are not necessary for understanding the sentence. So in order to shorten the length of the sentence as much as possible, we will remove stop words in this step.

E.g

The user's input is "hi, i want to know the lecturer address of comp5426."

The output is ['hi', 'want', 'know', 'lecturer', 'address', 'comp5426']

In this sentence we can identify some components that appear frequently but have no meaning to the whole sentence, such as “to” and “the”.

Figure 11 removing stop words

To ensure the stop words set can conclude all stop word, It is necessary to select a open source stop words package. For this project, we downloaded the NLTK stop word package.

Figure 12 download stop words

**2 Part of speech tagging**

POS refers to Part of speech tagging. The part-of-speech tagging is also a basic module in natural language processing, laying the foundation for syntactic analysis and information extraction. It is generally necessary to first segment the statement and then perform part-of-speech tagging.

The part-of-speech tagging algorithm is also divided into two categories, a dictionary search algorithm based on string matching and a statistical-based algorithm. The jieba participle combines two algorithms. For the words identified after the word segmentation, the word part is directly searched from the dictionary. For unregistered words, the HMM hidden Markov model and the viterbi algorithm are used to identify.

In Cassandra project, POS does not have the necessary role in the current structure. We mainly use POS for text analysis. In the later answer generation and the next stage of the raw question generation we may perform chunk parsing and dependency parsing on the text.

We introducing the NLTK(Natural Language Toolkit) to finish this step. It is necessary to download two helper package for POS.

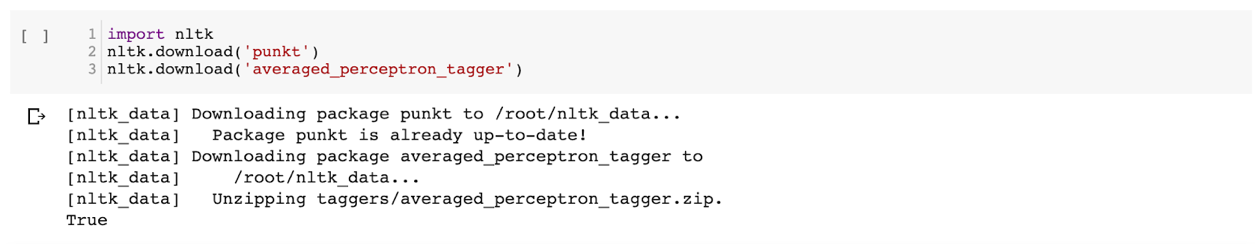


Figure 13 download POS package

And the main part of POS:

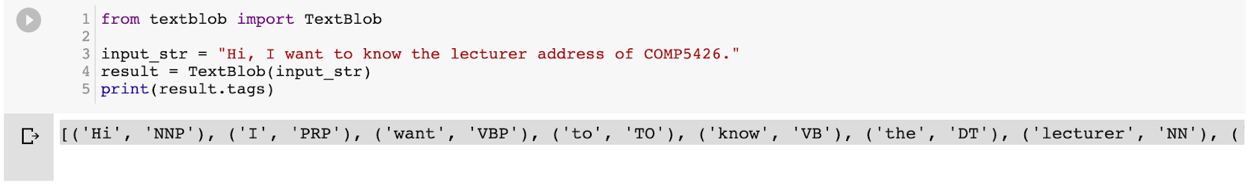


Figure 14 POS example

**3 Word Embedding**

Word embedding is a method of expressing words better by using a vector method. This method turns words into vectors that can justify the relationship between words. The main word embedding methods include word2vec and FastText.

Word embedding is a necessary step before trying to use intent classification. The easiest way to use word embedding is to give each word a label, or use one-hot encoding to turn a word into a vector, but none of these methods can reflect the relationship between words, so we intend to use the word2vec or fastText method.

图片包含 屏幕截图

描述已自动生成

Figure 15 FastText skipgram word embedding

图片包含 屏幕截图

描述已自动生成

Figure 16 Word2Vec skip-gram word embedding

After word2vec on the problem set, we found that the results were not available in our model because the number of words in our training set was too small. Therefore, we decided to use an open source word vector library. For the specific course name, the teacher name data is solved by the stream matching method.

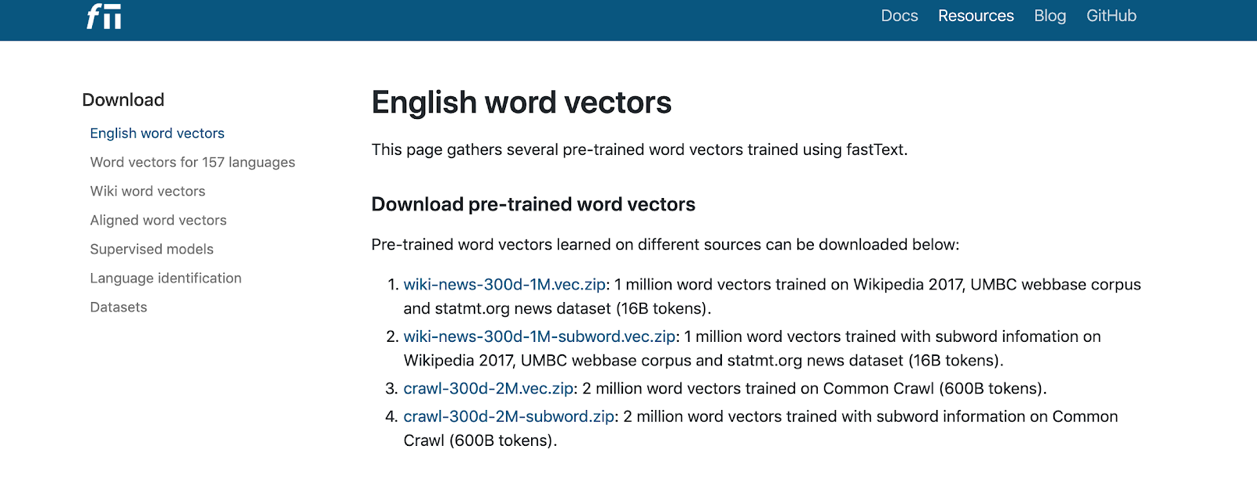


Figure 17 download wiki-news-300 word vector

图片包含 屏幕截图

描述已自动生成

Figure 18 example of load word vector

# Group Collaboration

图片包含 屏幕截图

描述已自动生成