Imouto NLP Documentation

INFR 4320U – Artificial Intelligence for Gaming

Kevin Pang

Kenny Difei Wang

Contents

# Summary

The Imouto NLP program is an experimental framework that integrates a soft Natural Language Processing component with a Finite State Machine in order to create an environment where the user can experience having an authentic conversation.

The program does not use a standard parser or dictionary; instead it uses a custom built parsing method and dictionary to process text. As a result, there are some limitations to the processing ability and understanding of the program.

# Program Components

The Imouto NLP program features the following main components.

1. TextProcessing.cs

TextProcessing.cs contains the methods that make up the core of the language processing component of the program. It processes text using text processing modules derived from the ThreadedTextProcessing class. As the name of the class suggests, the text processing modules run on their own threads.

Each module contains a list of words that it will parse the entered text for. For example, the positive keyword module would contain a list of words considered as positive keywords, while a swear word module would contain a list of expletives. If a matching word is found, it is logged onto a separate list called MatchedWords. The MatchedWords list is cleared at the start of each processing loop in order to eliminate the chance of overlapping results.

Once the modules are finished scanning the entered text, the program moves on to calculate a “sway value”. This value is used to determine state changes within the ImoutoFSM system. The formula for calculating the sway

1. ImoutoStates.cs
2. ImoutoObject.cs
3. ImoutoFSM.cs
4. FSMController.cs