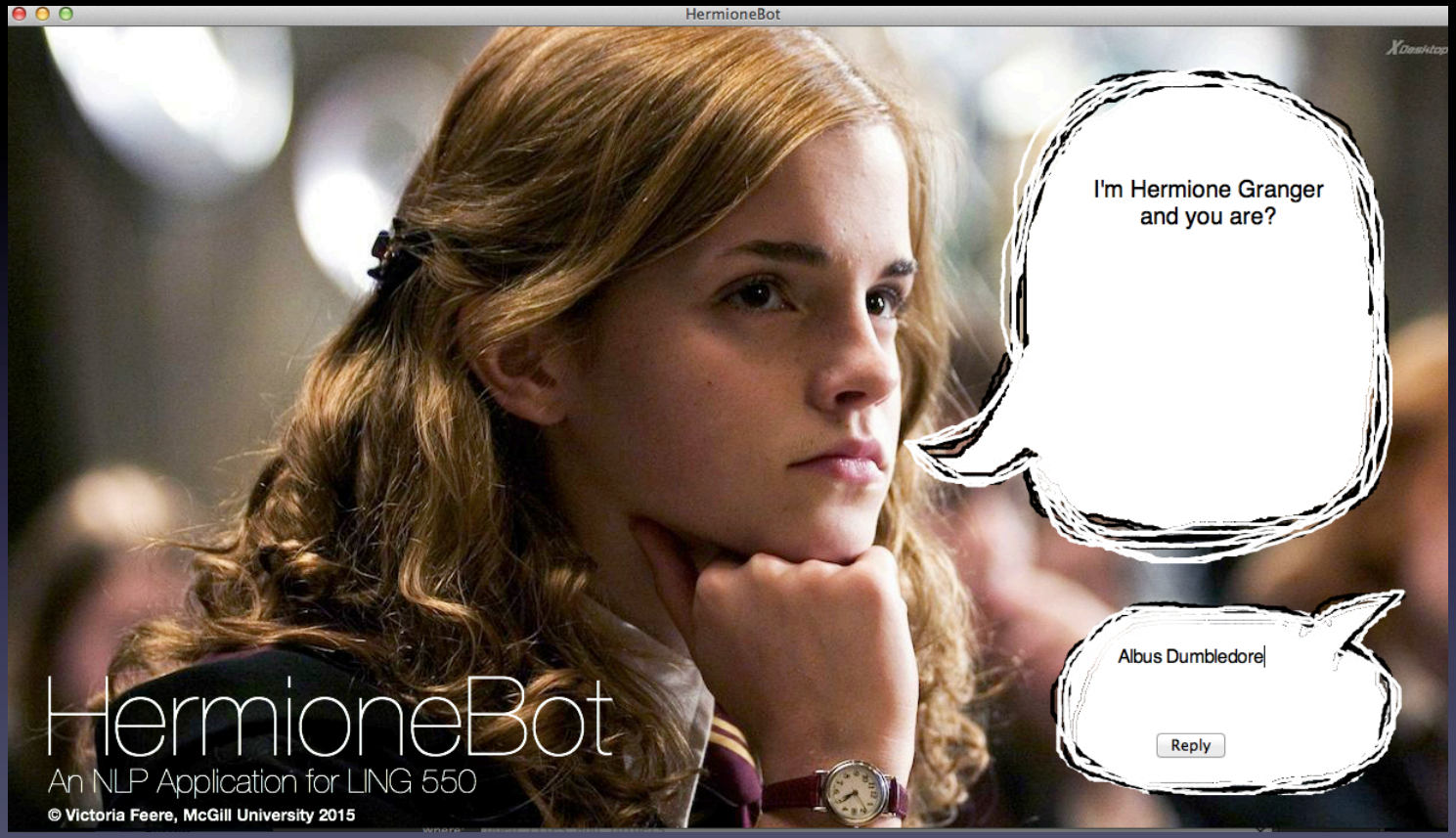
# HermioneBot




A Harry Potter trivia chatbot using nltk

# Motivation & Brief History

# Wikia API

- An API used to grab the content of articles on the Harry Potter wiki as json
- url: http://www.harrypotter.wikia.com/api/v1
- Used various encoded URI fragments to search and retrieve article content

# Understanding Queries

- Tokenized input using nltk's Punkt Tokenizer Models and word_tokenize(*userInput*)
  - **Example**:

'What is Quidditch?' -> ['What', 'is', 'Quidditch']

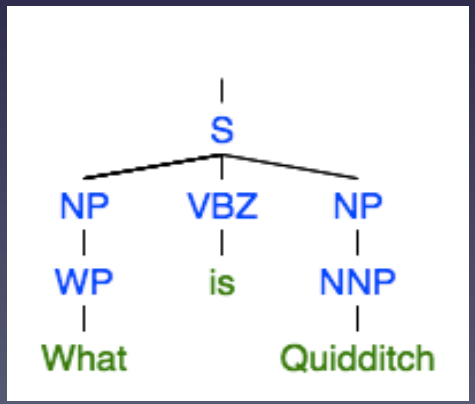
# Understanding Queries

- Performed POS-tagging on the query using Treebank POS Tagger and pos_tag(*tokenizedInput*)
  - **Example**: [‘What’, ‘is’, ‘Quidditch’] ->

    [(‘What’, ‘WP’), (‘is’, ‘VBZ’,), (‘Quidditch, ‘NNP’)]

# Understanding Queries

- Analyzed sentence structure using a regex parser generated using a specified grammar for NP/VP/PP-chunking
  - Subset of rules for NP-chunking:
    - grammar = "NP: {<DT>?<JJ>*<NN|NNS>+}"
  - parser = RegExpParser(*grammar*)
- Parsed query:
  - parser.parse(*taggedInput*)



# Understanding Queries

- Strategy: Identify queries based on sentence structure
  1. (S (WH-NP) (VP))
     - Who was Albus Dumbledore?
     - Where is Hogwarts located?
     - Who is the love of Hermione's life?
     - When did Bellatrix Lastrange escape from Azkaban?
  2. (S (NP) (VP … (WH-NP)))
     - The spell Wingardium Leviosa does what?
  3. (S (Aux) (NP) (VP| AdjP))
     - Does every spell require a wand?
     - Are you friends with Harry Potter?

# HermioneBot Algorithm

1. Using Tkinter, the built-in python GUI package, generate text bubbles and prompt the user for their name.
2. Store the user's name and allow user to ask questions.
3. On receiving input determine the user's intent based on syntactic structure of input:
   - Intents = {QUERY, UNKNOWN}
   - If QUERY:
     - Perform spell check, if spelling error detected return suggestion
     - Given parse, select primary queries (simplified NP) and additional keywords (verbs)
     - Query wikia for articles relating to queries
     - Iterate through article text predicting relevance given queries and keywords
     - Return most relevant 1 – 2 sentences
   - If UNKNOWN or empty – return sassy quip, Hermione style

# Spell Checking

- Before searching & scanning articles on wikia perform spell check
- Calculate edit-distance against pre-defined vocabulary
  - Vocab consists of spells, characters, etc. (domain-specific)
  - Uses *insert*, *replace*, *transpose* & *delete*
  - if min edit-distance < length(word) , return suggestion

# Demo!