

Severus Piton Dialog Manager

Giacomo Grandi

January 12, 2023

1 Introduzione

In questa relazione viene illustrata l'implementazione in Python di un Dialog Manager che impersona il professore Severus Piton durante un'interrogazione riguardo a delle pozioni magiche.

La relazione si articola seguendo la struttura del codice, illustrando tutti i file utilizzati dal sistema durante l'interrogazione: `potion.py`, `potions_list.py`, `frame.py`, `analysis.py`, `nlg.py`, `dialog_manager.py`.

1.1 Overview

Il Dialog System implementato conversa in inglese, poiché le librerie utilizzate sono più ricche in lingua inglese. Viene impersonato il professor *Severus Snape*, traduzione in lingua originale del personaggio della saga di Harry Potter, professore del corso di Pozioni alla Scuola di Magia e Stregoneria di Hogwarts.

Il chatbot presenta due modalità di interazione con l'utente, come spiegato in §6:

- l'utente può rispondere vocalmente, in inglese, alle domande del professore
- oppure può rispondere scrivendo, sempre in inglese, le proprie risposte sul terminale

Durante tutta l'interrogazione il professor Snape ha il controllo: è lui a proporre o chiedere possibili ingredienti allo studente, il quale si limita a cercare di completare gli ingredienti della pozione corrente senza bocciare l'esame.

2 Potion

Le pozioni sono l'oggetto richiesto allo studente dal professore durante l'interrogazione. La classe che le rappresenta è molto semplice: esse hanno solamente un nome e degli ingredienti. Nel file `potions_list.py` ho inserito 3 diverse pozioni prese dal sito: https://harrypotter.fandom.com/wiki/List_of_potions.

Le pozioni scelte sono:

- Polyjuice potion

- Felix Felicis
- Healing potion

È presente inoltre una lista completa di tutti gli ingredienti delle varie pozioni.

3 Frame

Il sistema è frame-based, come suggerito dalle indicazioni dell'esercizio: ogni pozione oggetto di una determinata interrogazione è rappresentata da un frame.

All'interno del frame è presente la pozione *target* e una lista di *current ingredients*, inizialmente vuota, che si aggiorna via via durante la discussione se lo studente risponde correttamente alle domande.

L'aggiunta di un ingrediente al frame viene fatta ignorando il caso sia dell'ingrediente proposto che degli ingredienti target, in modo da non doversi preoccupare di maiuscole e minuscole.

Inoltre è presente un metodo per controllare se il frame è completo o meno, che sarà la condizione di successo dell'interrogazione.

4 Analysis

La classe `Analysis` si occupa dell'analisi del testo e viene utilizzata per il controllo sia del testo generato che di quello prodotto dall'utente.

Per l'analisi del testo ho utilizzato la libreria `spaCy`: <https://spacy.io/>. Per capirne meglio il funzionamento ho utilizzato la funzione `get_dependencies()`, che restituisce un dizionario della forma: `{token: (dependency, part of speech, head, morphology)}`.

Ad esempio per la frase *"No, you don't need fluxweed for this potion."* (dove *fluxweed* è un ingrediente delle pozioni), l'output sarà:

```

1 no: ('intj', 'INTJ', 'need', )
2 ,: ('punct', 'PUNCT', 'need', PunctType=Comm)
3 you: ('nsubj', 'PRON', 'need', Case=Nom|Person=2|PronType=Prs)
4 do: ('aux', 'AUX', 'need', Mood=Ind|Tense=Pres|VerbForm=Fin)
5 n't: ('neg', 'PART', 'need', Polarity=Neg)
6 need: ('ROOT', 'VERB', 'need', VerbForm=Inf)
7 fluxweed: ('dobj', 'NOUN', 'need', Number=Sing)
8 for: ('prep', 'ADP', 'need', )
9 this: ('det', 'DET', 'potion', Number=Sing|PronType=Dem)
10 potion: ('pobj', 'NOUN', 'for', Number=Sing)
11 .: ('punct', 'PUNCT', 'need', PunctType=Peri)

```

Listing 1: `get_dependencies()` output

In questo modo ho capito come interpretare le frasi negative e creare pertanto un metodo per riconoscerle. Con la funzione `check_positivity()` controllo per tutti i token della frase la feature morfologica *Polarity* e se ne trovo almeno una negativa restituisco *False*.

Dato che, come anche mostrato nell'esempio precedente, la parola *"no"* non viene identificata con polarità negativa, ho aggiunto questo controllo alla funzione: pertanto anche se il testo

conterrà solamente il token *no*, esso verrà interpretato come frase negativa.

Per quanto riguarda invece il controllo della presenza di un ingrediente nel testo, ho fatto le seguenti considerazioni.

Gli ingredienti target delle varie pozioni sono molto differenti tra loro e spesso composti da più parole. Alcuni esempi significativi:

- Dragon liver
- Shredded boomslang skin
- Powdered bicorn horn
- Knotgrass

L'output delle dipendenze di questi ingredienti scritti all'interno di una semplice frase come "*There is *ingredient* in this potion*", è il seguente:

```
1 there: ('expl', 'PRON', 'is', )
2 is: ('ROOT', 'VERB', 'is', Mood=Ind|Number=Sing|Person=3|Tense=Pres|
      VerbForm=Fin)
3 dragon: ('compound', 'NOUN', 'liver', Number=Sing)
4 liver: ('attr', 'NOUN', 'is', Number=Sing)
5 in: ('prep', 'ADP', 'is', )
6 this: ('det', 'DET', 'potion', Number=Sing|PronType=Dem)
7 potion: ('pobj', 'NOUN', 'in', Number=Sing)
```

Listing 2: Dragon liver

```
1 there: ('expl', 'PRON', 'is', )
2 is: ('ROOT', 'VERB', 'is', Mood=Ind|Number=Sing|Person=3|Tense=Pres|
      VerbForm=Fin)
3 shredded: ('amod', 'VERB', 'skin', Aspect=Perf|Tense=Past|VerbForm=Part)
4 boomslang: ('compound', 'PROPN', 'skin', Number=Sing)
5 skin: ('attr', 'NOUN', 'is', Number=Sing)
6 in: ('prep', 'ADP', 'skin', )
7 this: ('det', 'DET', 'potion', Number=Sing|PronType=Dem)
8 potion: ('pobj', 'NOUN', 'in', Number=Sing)
```

Listing 3: Shredded boomslang skin

```
1 there: ('expl', 'PRON', 'is', )
2 is: ('ROOT', 'VERB', 'is', Mood=Ind|Number=Sing|Person=3|Tense=Pres|
      VerbForm=Fin)
3 powdered: ('amod', 'ADJ', 'horn', Degree=Pos)
4 bicorn: ('amod', 'ADJ', 'horn', Degree=Pos)
5 horn: ('attr', 'NOUN', 'is', Number=Sing)
6 in: ('prep', 'ADP', 'is', )
7 this: ('det', 'DET', 'potion', Number=Sing|PronType=Dem)
8 potion: ('pobj', 'NOUN', 'in', Number=Sing)
```

Listing 4: Powdered bicorn horn

```

1 there: ('expl', 'PRON', 'is', )
2 is: ('ROOT', 'VERB', 'is', Mood=Ind|Number=Sing|Person=3|Tense=Pres|
      VerbForm=Fin)
3 knotgrass: ('attr', 'NOUN', 'is', Number=Sing)
4 in: ('prep', 'ADP', 'knotgrass', )
5 this: ('det', 'DET', 'potion', Number=Sing|PronType=Dem)
6 potion: ('pobj', 'NOUN', 'in', Number=Sing)

```

Listing 5: Knotgrass

Come si vede da questi esempi, `spaCy` attribuisce ai vari token degli ingredienti dipendenze e POS tags molto diversi tra loro. E questi cambiano ancora se il testo fosse solamente composto dalle parole degli ingredienti. Ad esempio se il testo fosse solo *"Dragon liver"*:

```

1 dragon: ('compound', 'PROPN', 'liver', Number=Sing)
2 liver: ('ROOT', 'PROPN', 'liver', Number=Sing)

```

Listing 6: Dragon liver come 'ROOT'

Pertanto per rendere il controllo della presenza degli ingredienti nel testo più semplice e meno macchinoso (senza dover quindi controllare per dipendenze e POS tags differenti) ho deciso di confrontare semplicemente il testo di ogni token (e di eventuali vicini, nel caso di ingredienti composti) con il testo degli ingredienti.

Infine è presente un metodo che ritorna il numero di ingredienti trovati nel testo; tale metodo, come spiegato meglio in §5, viene usato per controllare che le domande generate contengano solamente 1 ingrediente.

5 Natural Language Generation

Per la generazione del testo, implementata all'interno della classe `NLG`, ho utilizzato la libreria `markovify`: <https://pypi.org/project/markovify/>.

Citando la descrizione del progetto al link precedente, `Markovify` è un semplice generatore di *"Markov chains"*. Essenzialmente ogni parola del corpus è collegata ad ogni altra con probabilità diverse, tramite delle catene di Markov. Quindi considerando ad esempio la parola *"Do"* come parola iniziale, `Markovify` assegna a tutte le altre parole del corpus la probabilità di seguire tale parola. Potrei ad esempio avere *"you"* con una probabilità molto alta, determinata da frasi tipo: *"Do you think *ingredient* is in the potion?"*.

L'assegnamento di tali probabilità avviene durante l'allenamento del modello su un dato corpus.

In particolare nel mio progetto ho utilizzato 3 corpus differenti, prodotti manualmente:

- `questions.txt`
- `positive_answers.txt`
- `negative_answers.txt`

Utilizzati rispettivamente per allenare il mio modello per produrre delle domande, delle risposte positive e delle risposte negative.

Il corpus su cui allenare il modello viene passato come parametro durante la creazione di un

oggetto della classe `NLG`, pertanto nel `Dialog Manager` dovrò creare un `question generator`, un `positive answer generator` e un `negative answer generator`.

Inoltre il `question generator` farà uso del metodo `generate_question()` per generare appunto una domanda. Tale metodo utilizza la funzione `make_sentence()` di `Markovify` per generare una frase random a partire dal corpus. Ho inoltre settato il parametro `max_overlap_ratio` della funzione `make_sentence()` a 0.9. Con questo setup, scelto dopo averne provati vari, il mio modello cercherà di generare una frase che non si sovrapponga più del 90% a quelle del corpus. Eseguirà un massimo di 10 prove (valore di default) per creare una frase che rispetti tale condizione, in caso di fallimento viene restituito `None`. Ho quindi inserito un controllo: se trova `None` viene ritornata la stessa funzione per provare nuovamente.

A questo punto il testo generato viene analizzato con la classe `Analysis` e se vengono trovati zero o più di due ingredienti viene scartata e generata una possibile nuova domanda, finché non viene prodotta una domanda contenente esattamente un ingrediente.

Ho scelto questo approccio perché le frasi senza ingredienti generate non erano abbastanza corrette semanticamente da poter essere inserite all'interno di una conversazione volta alla richiesta di ingredienti target. Venivano prodotte frasi del tipo:

- In order to make this potion?
- Can I make the potion isn't it?
- Is it possible to make this potion?

Pertanto, come spiegherò in seguito in §6, il professore potrà o proporre un ingrediente in particolare, oppure sceglierà casualmente all'interno di un pool di domande per richiedere allo studente di proporre un ingrediente.

Per quanto riguarda invece la generazione di frasi di risposta, entrambi i `positive` e `negative answer generators` faranno uso del metodo `generate_answers()` che funziona ugualmente al precedente ma senza il controllo del numero di ingredienti.

Tuttavia le frasi, soprattutto le domande in quanto più complesse, non sono sempre di senso compiuto, poiché non viene fatto alcun tipo di ragionamento semantico. Ad esempio vengono prodotte frasi come:

- Would it be correct to say that the knotgrass is in the potion would that be correct?
- Is it possible to make the potion be made with an occamy eggshell?
- Can I make the potion, do I need murtlap tentacles?

Quindi, anche in base a quanto scritto in <https://pypi.org/project/markovify/> nella sezione *Extending markovify.Text*, ho provato a creare la classe `POSifiedText` per fare override dei metodi `word_split` e `word_join` della classe `markovify.Text`. Questa classe, che fa uso di `spaCy` per fare lo split delle parole, dovrebbe permettere di produrre un modello che generi frasi che rispettino meglio la struttura delle frasi in lingua inglese. Tuttavia il miglioramento è talmente lieve che ho preferito utilizzare il modello originale e lasciare la classe `POSifiedText` solamente a scopo dimostrativo.

6 Dialog Manager

La classe `DialogManager` è quella che gestisce per intero l'interrogazione portata avanti dal chatbot del professor Snape.

Viene selezionata una pozione a caso tra le tre presenti in `potions_list` e creato un frame corrispondente. Vengono salvati gli ingredienti target, settate le `wrong_answers` a 0 (numero di risposte sbagliate dallo studente), creati question, positive answer e negative answer generators e inizializzata la memoria del discorso come una lista vuota.

Viene inoltre inizializzato il motore vocale della libreria `pyttsx3`: <https://pypi.org/project/pyttsx3/>. Questa libreria mi permette di fare la conversione text-to-speech in modo immediato, lavorando offline e facendo uso delle voci sintetizzate già presenti nel mio computer.

Viene infine salvato il parametro `can_listen`, che se positivo permette l'interazione vocale con il chatbot, se invece negativo consente di interagire scrivendo sul terminale le proprie risposte.

6.1 Interview

Il metodo `interview()` è quello che manda avanti la discussione tra lo studente e il professore.

L'interrogazione si apre con un greeting da parte del professore, in cui si presenta e stabilisce la pozione su cui verterà l'interrogazione.

Dopodichè si entra in un ciclo `while` da cui si uscirà solamente se il frame sarà completato, ovvero lo studente verrà promosso, oppure se vengono commessi 4 errori nelle risposte (`wrong_answers = 4`), e lo studente verrà quindi bocciato.

Ad ogni iterazione del ciclo viene scelta randomicamente una domanda da porre allo studente:

- con probabilità 40% verrà generata una domanda contenente un ingrediente tramite il question generator, tale domanda potrebbe essere veritiera oppure un trabocchetto (ovvero non contenere un ingrediente target)
- con probabilità 60% invece viene scelta a caso una domanda per chiedere allo studente che proponga lui un ingrediente target della pozione corrente

Se capita il caso di dover generare una domanda si controlla che la domanda generata non contenga un ingrediente presente in memoria, ovvero che era stato precedentemente detto durante il discorso (chiesto dal professore o proposto dallo studente).

In ogni caso l'interrogazione procede un ingrediente alla volta.

Dopo aver stampato e pronunciato la domanda, si aspetta la risposta dell'utente. La risposta viene analizzata e viene quindi generata la risposta corretta da far dire al chatbot in base al contesto che si è creato:

- Il professore chiede un ingrediente **target** e lo studente risponde **positivamente**: questo caso è **corretto**
- Il professore chiede un ingrediente **target** e lo studente risponde **negativamente**: questo caso è **sbagliato**

- Il professore chiede un ingrediente **trabocchetto** e lo studente risponde **positivamente**: questo caso è **sbagliato**
- Il professore chiede un ingrediente **trabocchetto** e lo studente risponde **negativamente**: questo caso è **corretto**
- Lo studente propone un ingrediente **target**: questo caso è **corretto**
- Lo studente propone un ingrediente **sbagliato** o **inesistente**: questo caso è **sbagliato**

In ogni caso quando un ingrediente viene menzionato, dallo studente o dal professore, esso viene salvato nella memoria del discorso.

Vi è infatti un ulteriore particolare caso di risposta dell'utente: cioè quando lo studente propone un ingrediente che è già tra i current ingredients. In questo caso non viene incrementato il numero di errori dello studente ed essenzialmente non succede nulla, se non che il professor Snape stuzzica lo studente dandogli dello "stupido", in perfetto stile del personaggio.

Quindi dopo aver controllato la risposta dello studente ed averne stabilito la correttezza, il chatbot genera una risposta appropriata utilizzando il positive answer generator o il negative answer generator. Dopodiché controlla che lo studente non abbia fatto troppi errori e poi prosegue con la prossima iterazione.

Se lo studente boccia Snape commenta il suo profondo disprezzo. Se invece riesce a completare la pozione target allora il professore calcola il voto e commenta adeguatamente al voto ottenuto.

Il calcolo del voto ottenuto parte da 31, considerato come 30 e lode, ed è dato dalla funzione: `math.floor(31 - wrong_answers * multiplier)`

Dove `multiplier` è scelto a caso tra `[3, 3.8, 4.3]`, per differenziare le votazioni (in questo modo si può ottenere voti tra il 18 e il 31). Il `multiplier` è indubbiamente una grossa penalizzazione, ma rispecchia il carattere del professor Snape come descritto nei libri, ovvero che si spazientisce velocemente con gli errori dei suoi studenti.

6.2 Text To Speech

Come già accennato precedentemente, ho implementato la funzionalità di text-to-speech nel mio progetto. Il chatbot del professor Snape infatti è dotato di una voce, in particolare l'unica voce sintetizzata di lingua inglese britannico e di sesso maschile presente sul mio computer, ovvero `com.apple.voice.compact.en-GB.Daniel`.

Tramite la libreria `pyttsx3` ho potuto impostare l'id della voce scelta e ho consentito al chatbot di pronunciare tutte le frasi che genera, ottenendo un'interazione più "umana".

6.3 Speech Recognition

Se l'oggetto della classe `DialogManager` viene creato con il parametro `can_listen = True` l'interazione con il professore sarà vocale e verrà quindi effettuato speech recognition.

Per questo scopo ho utilizzato la libreria `SpeechRecognition`: <https://pypi.org/project/SpeechRecognition/>.

Grazie a questa libreria posso impostare il microfono del mio computer come strumento di

ascolto e tramite la funzione `listen()` ascoltare effettivamente l'utente. È settata con un `timeout` di 5 secondi, in questo modo lo studente ha 5 secondi per pensare a una risposta prima che venga considerata come una risposta vuota.

Dopo aver ascoltato le parole dell'utente, con la funzione `recognize_google()` viene utilizzata la Google Speech Recognition API per interpretare ciò che lo studente ha detto.

Non ho creato delle chiavi personali per l'utilizzo di questa API di Google, ma sto utilizzando delle chiavi generiche che Google potrebbe revocarmi in ogni momento. Tuttavia dovrebbero poter essere utilizzate per test personali senza problemi.

Dopodiché viene stampato ciò che è stato capito delle parole dell'utente e analizzato alla ricerca di ingredienti target, proprio come una risposta scritta a terminale.

Con questa impostazione si ha l'interazione più "umana" possibile.

7 Esempi di interrogazioni

Verranno ora mostrati degli esempi di discussione tra il chatbot e lo studente.

```
1 Professor Snape: Hello, I am Severus Snape, the potions master. I will ask
   you about the ingredients of the Polyjuice Potion.
2     Should I put the leeches in the potion?
3 Student: Yes you should put them.
4 Professor Snape: That's correct, I am astonished, it is correct.
5     If I say that I need murtlap tentacles?
6 Student: No, that's a trick. You don't need them.
7 Professor Snape: Yes it is right, I knew a slytherin would get it right.
8     Can I make the potion is made with an occamy eggshell?
9 Student: No there is not the occamy eggshell.
10 Professor Snape: Yes this is the right answer.
11     Now I want to know if the shredded boomslang skin is in the potion?
12 Student: Yes there is!
13 Professor Snape: You really know your stuff, this is the right answer.
14     Can I put lacewing flies in the potion?
15 Student: Yes you have to put them.
16 Professor Snape: Yes it is right, I knew a slytherin would get it right.
17     Can you tell me an ingredient of the potion?
18 Student: There is knotgrass in this potion.
19 Professor Snape: Yes, it is correct.
20     Can I make the potion, do I need to put wormwood in the potion isn't
   it?
21 Student: No you don't need it.
22 Professor Snape: That's correct, I am truly amazed, it is correct.
23     Can you tell me an ingredient of the potion?
24 Student: Another ingredient is the fluxweed.
25 Professor Snape: Yes, it is correct.
26     Is the tincture of thyme, What do you tell me?
27 Student: No there is not this ingredient.
28 Professor Snape: Yes, it is correct.
29     Is it possible to put a squill bulb in the potion?
30 Student: No you don't put it in the polyjuice potion.
31 Professor Snape: Yes, it is correct.
32     Do you believe that there is powdered bicorn horn in the potion.
33 Student: Yes I believe so.
```


34 Professor Snape: That's correct, I am astonished, it is correct.
 35 Do you believe that there is a hair in the potion?
 36 Student: Yes there is also a hair.
 37 Professor Snape: That's correct, I am astonished, it is correct.
 38 You definitely appreciate the exact art of potion making, my dear
 student. You possess the predisposition to become one of the best
 potion makers in the world.
 39 I am very proud of you, your grade is 30 cum laude.

Listing 7: Interrogazione perfetta

1 Professor Snape: Hello, I am Severus Snape, the potions master. I will ask
 you about the ingredients of the Healing Potion.
 2 What do you need to make this potion?
 3 Student: I might need a hair.
 4 Professor Snape: This is not correct.
 5 What do you need to make this potion?
 6 Student: Maybe dittany.
 7 Professor Snape: That's correct, I am truly amazed, it is right, I knew a
 slytherin would get it right.
 8 Can I put lacewing flies in the potion?
 9 Student: No you can't.
 10 Professor Snape: Yes this is correct.
 11 Can I make the potion with an occamy eggshell?
 12 Student: No, again you can't.
 13 Professor Snape: Yes it is right, I knew a slytherin would get it right.
 14 Do you know an ingredient of the Healing Potion?
 15 Student: I can say that you need wormwood.
 16 Professor Snape: Yes this is the right answer.
 17 Which might be an ingredient of this potion?
 18 Student: Another ingredient is the bubotuber pus.
 19 Professor Snape: Yes, it is correct.
 20 Which ingredient do you think is in the Healing Potion?
 21 Student: I think there is the dragon liver.
 22 Professor Snape: That is the right answer my student.
 23 Which ingredient do you think is in the Healing Potion?
 24 Student: Lastly there is the unicorn tailhair.
 25 Professor Snape: That is the right answer my student.
 26 My student, indeed you are a great potion maker. Your potential is
 interesting.
 27 You passed the exam with a grade of 27.

Listing 8: Interrogazione con 1 errore

1 Professor Snape: Hello, I am Severus Snape, the potions master. I will ask
 you about the ingredients of the Felix Felicis.
 2 Do you know an ingredient of the Felix Felicis?
 3 Student: I think an ingredient might be leeches.
 4 Professor Snape: This is not correct.
 5 Do you know an ingredient of the Felix Felicis?
 6 Student: The ashwinder egg may be an ingredient.
 7 Professor Snape: I am truly amazed, it is right, I knew a slytherin would
 get it right.
 8 Can you tell me an ingredient of the potion?
 9 Student: There is powdered common rue in this potion

10 Professor Snape: Yes this is the right answer.
 11 Which ingredient do you think is in the Felix Felicis?
 12 Student: There is also the tincture of thyme.
 13 Professor Snape: That is the right answer my student.
 14 The unicorn tailhair to the potion?
 15 Student: Yes I think there is.
 16 Professor Snape: No this is not correct.
 17 Which might be an ingredient of this potion?
 18 Student: The occamy eggshell is an ingredient.
 19 Professor Snape: Yes this is the right answer.
 20 What about the fluxweed, is it in the potion?
 21 Student: No it isn't.
 22 Professor Snape: That's correct, I am astonished, it is correct.
 23 Is it possible to make the potion with bubotuber pus?
 24 Student: No it is not possible.
 25 Professor Snape: Yes, it is correct.
 26 Which ingredient do you think is in the Felix Felicis?
 27 Student: The squill bulb is in there.
 28 Professor Snape: Yes, it is correct.
 29 Do you think lacewing flies are in the potion isn't it?
 30 Student: No they are not in the potion.
 31 Professor Snape: That is the right answer my student.
 32 Would it be correct to say that the knotgrass is in the potion?
 33 Student: No it is not correct.
 34 Professor Snape: Yes, it is correct.
 35 Do you know an ingredient of the Felix Felicis?
 36 Student: The ashwinder egg.
 37 Professor Snape: It is correct, but you already said it, are you dumb?
 38 Do I need murtlap tentacles?
 39 Student: Yes!
 40 Professor Snape: Yes this is the right answer.
 41 You probably did not pay attention to the class, you still have a lot
 to learn. But you are not a really bad student.
 42 You somehow passed the exam with a grade of 25.

Listing 9: Interrogazione con 2 errori

1 Professor Snape: Hello, I am Severus Snape, the potions master. I will ask
 you about the ingredients of the Felix Felicis.
 2 Would it be correct to put a squill bulb in the potion?
 3 Student: Yes it is correct.
 4 Professor Snape: Yes this is correct.
 5 Which ingredient do you think is in the Felix Felicis?
 6 Student: I think there is a hair.
 7 Professor Snape: This is not correct, you dumb student.
 8 Is it necessary to put wormwood in the potion?
 9 Student: Yes it is.
 10 Professor Snape: You don't get the art of potion making.
 11 Is it possible to make the potion is made with an occamy eggshell?
 12 Student: Yes you should add it to the potion.
 13 Professor Snape: You really know your stuff, this is the right answer my
 student.
 14 What do you need to make this potion?
 15 Student: I need murtlap tentacles.

16 Professor Snape: You really know your stuff, this is the right answer.
 17 What do you need to make this potion?
 18 Student: I also need an ashwinder egg.
 19 Professor Snape: That's correct, I am astonished, it is correct.
 20 Is it correct to add unicorn tailhair is in the potion?
 21 Student: Yes it is.
 22 Professor Snape: You don't really get the art of potion making at all.
 23 Which might be an ingredient of this potion?
 24 Student: The tincture of thyme might be an ingredient.
 25 Professor Snape: You really know your stuff, this is the right answer my student.
 26 Should I put the leeches in the potion?
 27 Student: No you shouldn't.
 28 Professor Snape: That is the right answer my student.
 29 Can you tell me an ingredient of the potion?
 30 Student: I need the powdered common rue.
 31 Professor Snape: I am astonished, it is right, I knew a slytherin would get it right.
 32 You got 19. Only a griffindor would reply to my questions like you did. You did pass the exam, but I don't want to waste my time with you anymore. Go away!

Listing 10: Interrogazione con 3 errori

1 Professor Snape: Hello, I am Severus Snape, the potions master. I will ask you about the ingredients of the Polyjuice Potion.
 2 What do you need to make this potion?
 3 Student: I need murtlap tentacles.
 4 Professor Snape: You don't get the art of potion making.
 5 Might there be shredded boomslang skin is in the potion?
 6 Student: No they are not.
 7 Professor Snape: No, you must be a griffindor to think that.
 8 Can you tell me an ingredient of the potion?
 9 Student: I can say there is a squill bulb.
 10 Professor Snape: You don't get the art of potion making.
 11 What do you need to make this potion?
 12 Student: Maybe dittany.
 13 Professor Snape: Don't waste my time.
 14 You failed the exam and you are wasting my time. Get out of my sight!

Listing 11: Interrogazione bocciata

1 Professor Snape: Hello, I am Severus Snape, the potions master. I will ask you about the ingredients of the Polyjuice Potion.
 2 Is it possible to make the potion be made with dragon liver?
 3 ...ready to listen...
 4 Student: no it is not possible
 5 Professor Snape: I am astonished, it is right, I knew a slytherin would get it right.
 6 Do you think lacewing flies in the potion?
 7 ...ready to listen...
 8 Student: yes they are in the potion
 9 Professor Snape: That is the right answer my student.
 10 Is it possible to make the potion, do I need powdered common rue being in the potion?

11 ...ready to listen...

12 Student: no it is not in the ocean

13 Professor Snape: You really know your stuff, this is the right answer my student.

14 Can you tell me an ingredient of the potion?

15 ...ready to listen...

16 Student: an ingredient is a hair

17 Professor Snape: That's correct, I am astonished, it is correct.

18 What about the fluxweed, is it in the potion?

19 ...ready to listen...

20 Student: yes it is

21 Professor Snape: That's correct, I am truly amazed, it is correct.

22 Is it correct to add unicorn tailhair is in the potion?

23 ...ready to listen...

24 Student: no it is not correct

25 Professor Snape: That's correct, I am truly amazed, it is correct.

26 Do I need bubotuber pus to make this potion?

27 ...ready to listen...

28 Student: no you don't need it

29 Professor Snape: Yes, it is correct.

30 Which might be an ingredient of this potion?

31 ...ready to listen...

32 Student: the leeches

33 Professor Snape: Yes, it is correct.

34 Which ingredient do you think is in the Polyjuice Potion?

35 ...ready to listen...

36 Student: you should add shredded boomslang skin

37 Professor Snape: That is the right answer my student.

38 Is it necessary to put wormwood in the potion?

39 ...ready to listen...

40 Student: no it is not

41 Professor Snape: That is the right answer my student.

42 Can you tell me an ingredient of the potion?

43 ...ready to listen...

44 Student: you need to add knot grass

45 Professor Snape: No this is not correct.

46 Which might be an ingredient of this potion?

47 ...ready to listen...

48 Student: the powdered bicorn horn

49 Professor Snape: Yes, it is correct.

50 Which might be an ingredient of this potion?

51 ...ready to listen...

52 Student: you need to add knot grass

53 Professor Snape: You must be a griffindor.

54 If I say that the knotgrass is in the potion?

55 ...ready to listen...

56 Student: yes it is

57 Professor Snape: Yes this is correct.

58 You probably did not pay attention to the class, you still have a lot to learn. But you are not a really bad student.

59 You somehow passed the exam with a grade of 25.

Listing 12: Interazione vocale con il chatbot

8 Risultati e commenti

Dall'analisi delle conversazioni che si ottengono con il Dialog Manager allo stato attuale ho individuato quattro principali problematiche:

- **Generazione del linguaggio:** il linguaggio generato non è sempre di senso compiuto, ed è spesso ripetuto (nelle risposte). A tal proposito potrebbe essere utile arricchire i vari corpus e inserire dei controlli sintattici e semantici nelle frasi generate. Oppure affidarsi a dei modelli di deep learning come ad esempio GPT-3.

- **Coerenza nel discorso:** allo stato attuale il professore non mostra risposte molto coerenti tra loro, come si nota chiaramente nel commento finale dell'interrogazione con 3 errori (10). Qua si nota infatti che prima il professore è molto contento della risposta corretta, ma subito dopo si mostra molto seccato del risultato complessivo dello studente.

Per questo proposito potrebbe essere utile implementare e tenere traccia del mood corrente del professore, che cambia continuamente in base alle risposte che propone lo studente, per generare del testo coerente con quanto sta succedendo nell'interrogazione e con il sentimento del professore.

- **Speech Recognition degli ingredienti:** come si nota nell'esempio di interazione vocale (12) la comprensione vocale funziona quasi perfettamente per tutte le parole "normali". Si nota solamente un errore nella linea 12, quando il chatbot capisce: "*no it is not in the ocean*", mentre la parola corretta era chiaramente "*potion*". Forse pronunciata non correttamente da me in quel frangente, poiché invece nella linea 8 è stata riconosciuta correttamente.

Tuttavia per le parole degli ingredienti, che sono notoriamente in un inglese "fantasy" e spesso inventato, l'automatic speech recognition fatica a capire e formulare correttamente l'ingrediente pronunciato. Come si vede infatti nella fase finale dell'interrogazione quando provavo a dire di voler aggiungere l'ingrediente *knotgrass*, esso veniva interpretato come due parole separate e quindi considerato come un errore. Quando poi, fortunatamente, mi viene proposto dal professore rispondo affermativamente e riesco così a completare gli ingredienti della pozione corrente, cosa che altrimenti sarebbe stato pressoché impossibile.

Per migliorare questa situazione forse, ma non so se sia possibile, si dovrebbe inserire la lista degli ingredienti delle pozioni e le pronunce relative nel dataset su cui si è allenato il motore di riconoscimento vocale.

- **Ricerca degli ingredienti nel testo:** attualmente un ingrediente viene trovato nel testo solamente se è scritto correttamente ogni suo singolo carattere. Potrebbe quindi essere un miglioramento inserire una qualche correzione automatica del testo, prima di essere analizzato, per salvarsi da errori di battitura e anche per aiutare il riconoscimento vocale degli ingredienti.