Language Learning Chatbot for Coptic

Sichang Tu
Department of Linguistics
Georgetown University
st1018@georgetown.edu

Abstract

Chatbots for language learning has been robust recently. However, the language learning chatbots for low-resourced languages have not been studied in depth. In this project, we dabble in the development of the language learning chatbot for Coptic, a historical Afro-Asiatic language in Egypt. Based on the Rasa framework, we present a system which is able to answer questions about Coptic culture, lexicon lookup and translation of English words. The statistics of model evaluation gives a more accurate description of the system, showing the system performs well in handling above tasks, and is a promising start point for the future development.

1 Introduction

Though the idea of developing chatbots that helps users to acquire a foreign language is not novel, most of the popular bots that people are familiar with, including Duolingo and Mondly, only cover prevailing languages. None of them targeted low-resourced languages such as Coptic. One major reason lies in the fact that the user group is restricted and the data is insufficient to train the model. The limited resources also pose challenges for those language learners. Taking Coptic as an example, most learning materials people can find online are books or translation of manuscripts that are written decades ago, which require learners to devote significant time and energy to mastering it. Even if there are few digital resources like online corpora and database, it is difficult for a non-expert to maneuver them. Therefore, building a language learning chatbot not only makes up holes in this filed, but also benefits potential language learners, providing an alternative, interesting and interactive way to acquire Coptic knowledge.

This system is currently using English as the major language, taking Coptic input according to the user's choice of question types. It is designed to help English speakers to quickly grasp the basic understanding of Coptic language and lexicons. To build the system, we adopt Rasa as the framework, since it has configurable pipeline to process the NLU and dialogue models and provides developers with high degree of freedom to customize the actions achieved by the system.

In this paper, we present a novel language learning chatbot for Coptic, which has never been investigated in previous work (Section 2). And we reported the design (Section 3) and a series of experiments (Section 4), including success and failure, in developing this system.

2 Related Work

Chatbots for educational purposes including language learning is an emerging research filed, which has drawn attention since the last decade (Fryer and Carpenter, 2006; Han, 2012). Previous work has almost targeted languages with large-scale data which enables the model to learn and achieve comparatively remarkable performance. Systems like The Tactical Language and Culture Training System (TLCTS) are designed to help users acquire communication skills and cultural knowledge in foreign languages (Johnson, 2007). Wik and Hjalmarsson introduced two systems, Ville and DEAL. The former acts as a virtual teacher for language and pronunciation training, while the latter enables users to obtain conversational skills in role-play games. Lopes et al. (2017) presented a social bot in a conversational setting, which aims to assist learners to practice Sweden according to their language levels. In recent years, chatbots for language learning has swarmed into the industry. One outstanding example is Tutor Mike, the winner in International Loebner Prize 2018 Selection Contest. It also inspired the development of this system.

Current studies and resources about Coptic are focusing on the following three areas:

- **Coptic corpora and digital resources** The most popular existing Coptic corpora and database is part of the Coptic Scriptorium project (https://github.com/CopticScriptorium/corpora). It provides Coptic text for reading, analysis and complex research in different formats, including TEI XML, PAULA XML, ConLL and ANNIS.

- **Coptic lexicon and dictionary** There are few projects which contributed to the establishment of Coptic dictionary, such as Coptic Scriptorium (https://copticscriptorium.org), CoptOT (http://coptot.manuscriptroom.com), and KELLIA (http://kellia.uni-goettingen.de). The lexicon preparation of the dictionary contains two parts: the *BBAW Lexicon of Coptic Egyptian* (https://aaew.bbaw.de/tla/) and the *DDGLC Lexicon of Greek Loanwords in Coptic* (https://www.geschkult.fu-berlin.de/en/e/ddglc).

- **NLP tools for Coptic** Besides corpora and dictionary, NLP tools like entity visualization, part-of-speech (POS) tagging and annotation tools (normalizer and lemmatizer) have been developed for Coptic. Zeldes and Schroeder (2016) have proposed an end-to-end open source tool chain to process Coptic text.

The development of language learning chatbot has not involved Coptic yet and the research for Coptic has not stepped into the field of dialogue system. Therefore, this project could benefit both sides since it bridges the two research areas. And thanks to the Coptic resource listed above, the implementations like lexicon search in the system can fully make use of those existing dictionary and NLP tools.

3 System Design

The framework used to build the system is *RASA version 2.0*, since it has flexible options to support customizing NLU pipelines, dialogue model and external API calls. In order to properly develop the system, we first craft the workflow of bot (see Figure 1).

As shown in the flow chart, the bot will greet and introduce itself as a Coptic tutor, and remember the user's information like their names if provided. Then the bot will offer 4 different question types for users to choose from. *Culture* deals with users' input like *Tell me more about Coptic.* And *dictionary* handles questions including looking up Coptic words and return their meaning, POS tag and morphological information. For example, if the users want to learn about a Coptic word, they can choose the 'dictionary'. Once the user chooses the type, the bot will return utterances like *What lexicon would you like to look up?* And the bot will process the sentence or words that the user inputs using the custom actions in the Rasa frame and call external APIs to look up the words, and return the search results. *Translation* is similar to *Dictionary*. It looks up the English words that the user inputs and return the corresponding Coptic words. *Comparison* refers to the comparison of two Coptic words. Users can ask questions like *What the difference between* ⲙⲛⲟϣⲓ *and* ⲙⲛⲟϣⲉ*?* If the user is satisfied with the results and does not ask further questions or make another query, the conversation will end. If the user would like to ask other questions, the system will repeat the whole process from the first decision node.

To achieve the four functions in the flow chart, we divide the whole task into following steps, which will be illustrated in detail in the next section: (1) craft adequate NLU data for each type of intents according to Rasa guideline; (2) identify question types at the first decision node; (3) differentiate Coptic text from English text; (4) create custom actions for dictionary, translation and comparison; (5) manage story path and configure the NLU and dialogue models.

4 Experiments

4.1 NLU Data Generation

Rasa requires developers to craft NLU data in the formats of *intents* (possible input from users) and *responses* (the responses from the chatbot) so that the model could learn the pattern and make predictions to the user's input. If there are any entities the system should pay attention to during training, they should be properly labeled in the NLU data. Therefore, we generate intents for each type of questions in the following format, taking *Comparison* as an example.

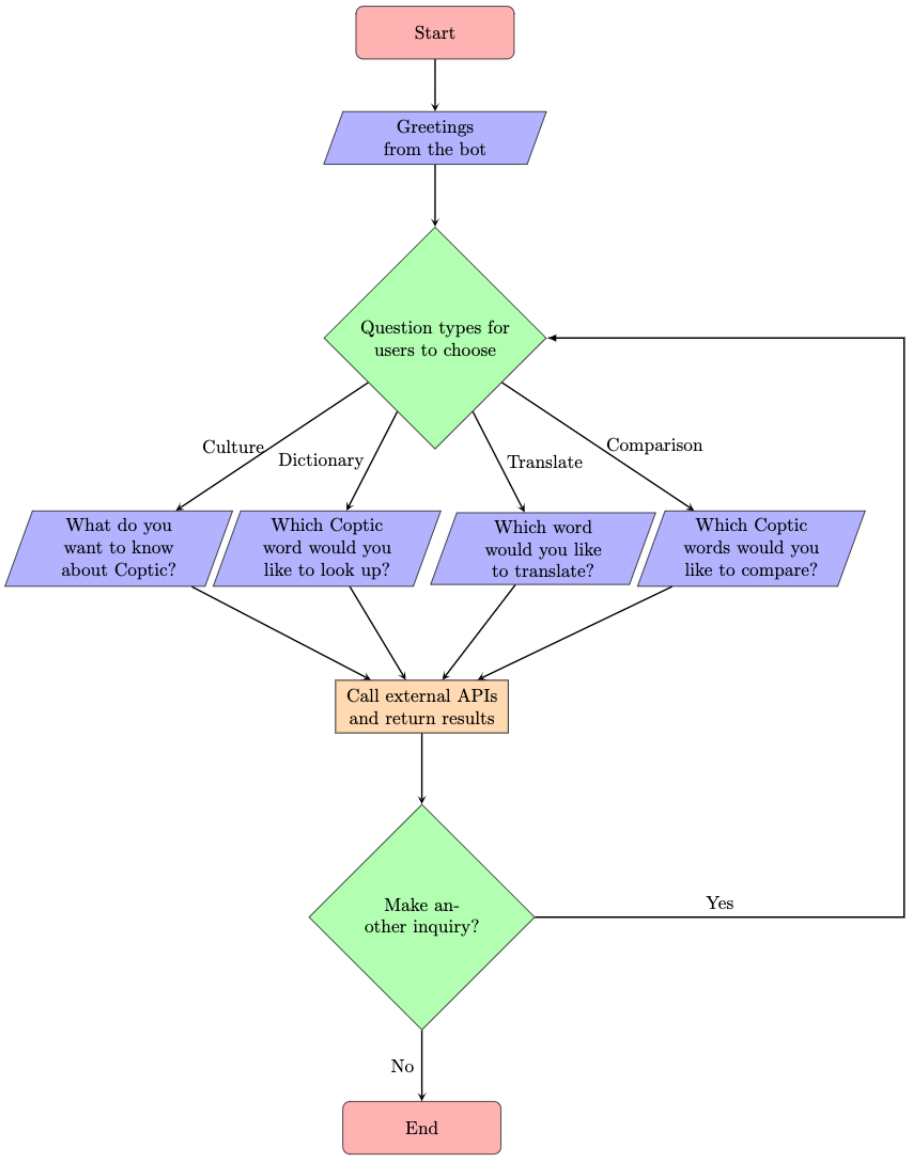


Figure 1: The designed work flow of this system.

(1) What is the difference between [ⲡⲉ] {"entity": "compare_1" } and ⲧⲉ {"entity": "compare_2" } ?

In example (1), the first Coptic word is labeled as *compare_1* and the second as *compare_2*. Thus the system will be able to recognize the which two words to compare with the similar sentence structure. In addition, for these entities, we set slots with the same names. Slots act as the memory of the bot, which will be auto-filled once the system detects the entities that have the same names as the slots. It makes it possible for the system to store important information extracted from users' input, such as their names. And these stored value can be freely called by the responses and custom actions.

4.2 Question Type Identification

One important step in the workflow of the chatbot is to determine which question type the user intends to ask. At first, we attempted to generate distinct NLU data for each type and let the system to learn the pattern. However, during practice, we found that the prediction accuracy of this method is low. The system always makes wrong decisions and thus result in the following crash of the bot. And if the system is indecisive about the question type, it will execute the default fall back action and returns no response.

In order to solve this problem, we decide to provide more guidance for the user in this step by adding option buttons. Users can choose from the 4 options. Once the user clicks on the buttons, the

system will automatically return the corresponding intent and continue the next action.

4.3 Coptic Language Detection

Another significant step in the workflow is the identification of Coptic language. For instance, if the user chooses the *Dictionary* and inputs *look up* ⲁⲅⲁⲡⲏ, the system need to first identify there is a Coptic word in this sentence and then extract the word. We tried to label the Coptic words in NLU data as entities and set slots. It nevertheless raised user warning about misaligned entity annotation during training. And test results also show that it has low slot accuracy. Hence, we later write a custom function to filter the Coptic text, which is not that challenging since Coptic has a distinguished range of unicode. The detection process in current system first obtains the whole sentence of users input via Rasa build-in *tracker* function and then run the filter function to extract the Coptic text.

4.4 Custom Actions

Apart from the filter function illustrated in section 4.3, we defined another three functions for dictionary, translation and comparison and one class for scraping lexicon information from external online Coptic dictionary and corpus.

Web Scraping Our first attempt to acquire the lexicon information is incorporating database of the online dictionary into the system. Unfortunately, this method is infeasible due to the fact that the dictionary do not solely rely on the database, but other external digital resources. Hence, we turned to the web scraping method, using python libraries *requests* and *beautifulsoup*. The challenges of scraping the website is that URL of the lexicon search results is dynamic. If the lexicon have multiple entries, we can easily parse the page through modifying one parameter in the URL. However, if the lexicon have only one entry, the URL will contain the unique lexicon ID returned by javascript function on the website instead of following the same pattern of the lexicon with multiple entries. The lexicon IDs are stored in the database, therefore we add anther function in this class to search for the IDs if the lexicon only has one entry. Through web scraping, the system is able to obtain the meaning, POS tags and morphological information of the lexicon.

Dictionary The dictionary function allows the system to look up any Coptic words that the user inputs. It first tracks the current utterance from the user, then filters out English text and calls *cop_to_eng* function to return the search results.

Translation Similar to the dictionary function, the translation function allows users to obtain corresponding Coptic words by inputting English words. This function was part of the dictionary function, but we found that it would be difficult for the system to recognize which English words to look up if they are combined. Thus, we decide to treat it as an independent function.

Comparison The detection of the two Coptic words in user's input utilizes both filter function and entities setting. One problem of achieving this function is the system always returns *TimeoutError* since it takes longer time to scraping the information. Therefore, we modified the core module of the Rasa framework to allow the system have more time to process the data.

5 Evaluation

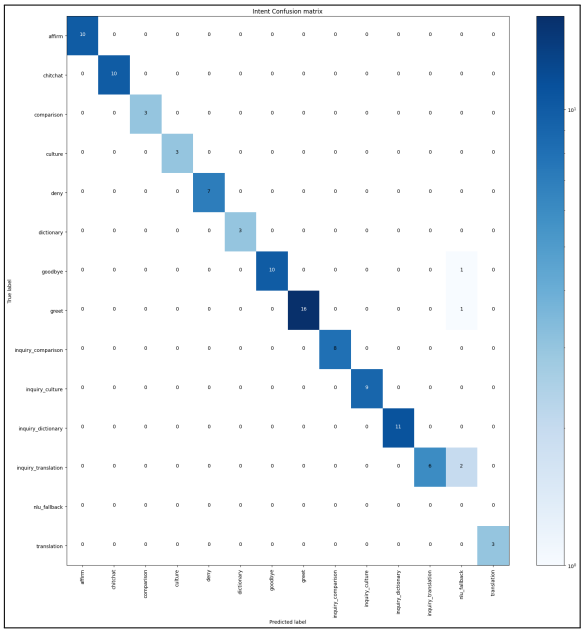
As for evaluation, we designed both objective metrics and subjective metrics to assess the system. Rasa provides test functions for developers to evaluate the model in terms of the percentage of task/actions complete rate, correctness rate of intents and entity prediction. Table 1 gives the evaluation on action level, which shows that the overall performance of the system regarding the action complete rate is promising.

	Correct	**F1 score**	**Precision**	**Accuracy**
action	14/16	0.920	0.920	0.920

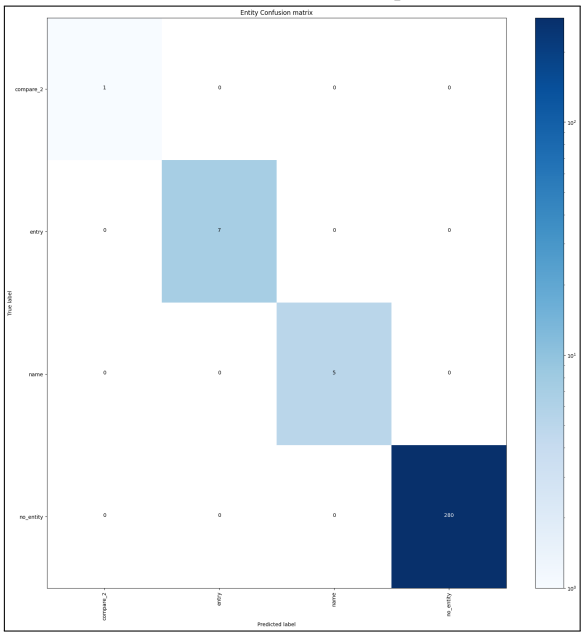
Table 1: Evaluation results on ACTION level.

Figure 2 provides the confusion matrices of intent prediction and entity classifier. As shown in Figure 2a, the general prediction of the intent are accurate, except for the *inquiry_translation*, which may be due to the insufficient training data of the intent type so that the system cannot identify this type well. This is also the reason why the system sometimes fail to make response to user's input in the question type *translation*. The Figure 2b indicates that the accuracy of entity classification is high, since there are only few types of entities for the system to identify.

Since the experience of the user is significant for developing the chatbot, we designed a survey form which is attached to the end of the work flow in



(a) Confusion matrix for intents prediction.



(b) Confusion matrix for entity classifier.

Figure 2: Confusion matrices for intents prediction and entity classifier.

the form of URL [1] to conduct subjective metrics. It contains 5 questions which we consider as the most important to ask:

1. On a 1 to 5 point scale, how would you rate this system?

2. What do you find most amazing about the system?

3. What do you find most frustrating about the system?

4. Would you recommend the system to other people?

5. Any other feedback you would like to share?

[1]The form can be found on: https://forms.gle/SDd7Fh2Usu8fdK1x8

6 Conclusions and Future Work

Based on the framework of Rasa, we have developed a language learning chatbot for Coptic, which is able to handle basic culture, dictionary, translation and comparison questions from the user. The overall performance of the system is quite promising except for *translation* questions.

The future work of this system will focus on reducing the current errors and incorporating more functions such as syntactic process of Coptic sentences. Due to the time limitation and long time of text processing time using current online tools, it is challenging to add this features into the system. In addition, we plan to apply the system as a website widget in the following experiments so that users do not have to download the chatbot.

References

Luke Fryer and Rollo Carpenter. 2006. Bots as language learning tools. *Language Learning & Technology*, 10(3):8–14.

Jeonghye Han. 2012. Robot assisted language learning. *Language Learning & Technology*, 16(3):1–9.

W. Lewis Johnson. 2007. Serious use of a serious game for language learning. In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, page 67–74, NLD. IOS Press.

José Lopes, Olov Engwall, and Gabriel Skantze. 2017. A first visit to the robot language café. In *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, pages 7–12.

Amir Zeldes and Caroline T. Schroeder. 2016. An NLP pipeline for Coptic. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 146–155, Berlin, Germany. Association for Computational Linguistics.