Mini-Project Three

Allen Worthley

mworthley3@gatech.edu

Abstract—This document discusses the design of an AI agent that extracts meaning from sentences given a list of common words. The agent interprets questions about the sentence and attempts to answer these questions.

1 DESIGN

The AI agent uses a frame representation of several primitive actions to extract meaning from and answer questions on given sentences. However, the bulk of the agent relies on encoding or categorizing a list of common words. The agent employs the encoded list to create heuristics that infer meaning objects from the encoded sentence text. The agent then interprets the question and either returns the identified meaning object or returns a guess.

1.1 Pre-processing

To categorize or tokenize words into parts of speech, the library spacy does some initial processing on the list of common words. However, the library is not entirely accurate. For example, the library categorizes the word "walk" as a noun which in some instances is correct. But, in the context of "David and Lucy walk to school", the part of speech associated with the word "walk" should be classified as a verb.

The design of the tokenized common word list relies on manual categorization to encode the appropriate part of speech to each word. While automatization is preferred, the agent is unable to infer what part of speech should be used for a given word based on the context.

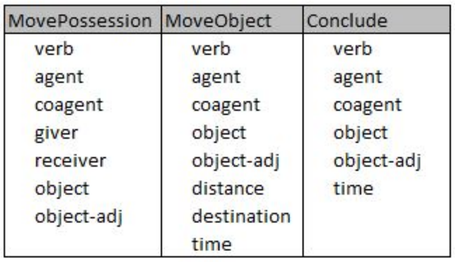
Additionally, certain nouns need further encoding. Nouns can represent categories of adjectives or other nouns. Nouns like “color” may pertain to adjectives like ‘blue’ or ‘green’. Also, nouns may lie in a category such as nouns related to distance, quantities, or measurements. For example, “many” alludes to a quantity of other nouns in the context of a question.

1.2 Data Structures

Dictionaries house the majority of the data where the key is the common word and the value is the category. Dictionaries exist to relate words to parts of speech, primitive actions, and special categories of nouns. In addition, for use in frames, lists contain parsed words from sentences and questions. An annotated nested list houses the original list of words and the associated parts of speech.

1.3 Frame Representation

The agent constructs three naive primitive action frames as classes: MoveObject, MovePossession, and Conclude. Within the frames, meaning objects collect the specific information from the sentence like agent, coagent, or distance. Figure 1 shows the full list of meaning objects for each frame. The AI agent uses a host of heuristics to extract information into these objects as discussed in the proceeding sections. These heuristics are generalizations and are rather simple.



MovePossession	MoveObject	Conclude
verb	verb	verb
agent	agent	agent
coagent	coagent	coagent
giver	object	object
receiver	object-adj	object-adj
object	distance	time
object-adj	destination	
	time	

Figure 1— Overview of meaning objects.

1.3.1 *Agent, Coagent, Giver, and Receiver*

The Agent is the first pronoun or proper noun within a sentence and Coagent is the second. If the primitive action associated with the verb is MovePossession and the sentence contains the preposition “to” or “from”, then the Giver and Receiver change. If “to” is in the sentence, then Agent is the Giver and Coagent is the Receiver. Otherwise, if “from” is in the sentence then the opposite is true.

1.3.2 *Object and Object-Adj*

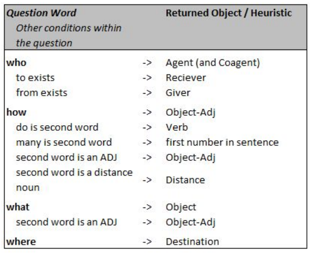
The Object is the first noun used after the verb. The Object-Adj is the adjective before the noun. If the primitive action for the verb is Conclude, then Object-Adj is just the first adjective after the verb.

1.3.3 *Distance, Destination, and Time*

The agent populates the Distance object with nouns and adjectives classified as distance in the noun category dictionary. Similarly, the agent populates the nouns associated with time in the Time object. Also, for time codes specifically, the agent tokenizes this as a time as a part of speech. If a time code is present within the sentence, the agent populates Time with the time code. For the MoveObject primitive action, Destination is the noun after the preposition “to”.

1.4 Question Interpretation

The AI agent also uses heuristics to interpret the questions. Figure 2 shows an overview of the mapping of the question conditions to the returned object.



Question Word *Other conditions within the question*		Returned Object / Heuristic
who	->	Agent (and Coagent)
to exists	->	Reciever
from exists	->	Giver
how	->	Object-Adj
do is second word	->	Verb
many is second word	->	first number in sentence
second word is an ADJ	->	Object-Adj
second word is a distance noun	->	Distance
what	->	Object
second word is an ADJ	->	Object-Adj
where	->	Destination

Figure 2— Overview of question heuristics.

2 PERFORMANCE

2.1 Weak points

The agent is naive and struggles with verbs that do not fit naturally within the three primitive actions. For example, the verb 'told' classified as the Conclude primitive action does not make much sense. However, the agent is able to answer 14 of the questions correctly with these simple heuristics.

2.2 Efficiency

The code is efficient as it runs in linear time to the number of words in the sentence and question. The most complicated portion of the code runs a for-loop to iterate over the words.

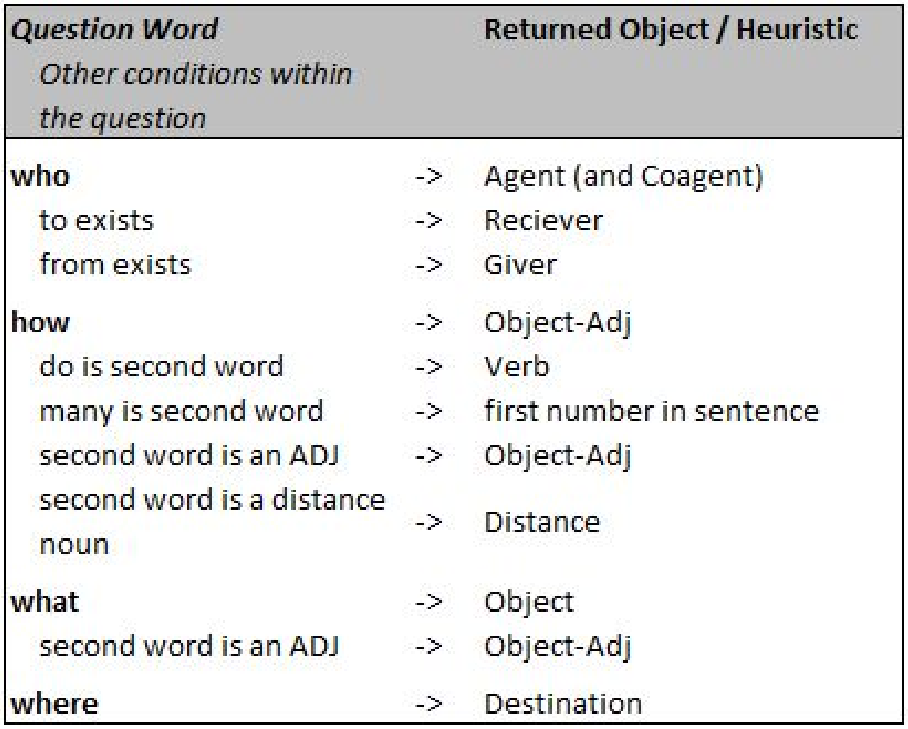
2.3 No Cleverness Here

The agent naively returns meaning objects based on heuristics and is not very clever.

2.4 Does the AI Agent Act Like a Human

Some of the heuristics the agent employs mimics how a human may subconsciously interpret a sentence and question. For example, the AI agent interpreting the Receiver or Giver of an object follows how a human thinks. The agent changes the values of these objects depending on the preposition. However, other aspects of this AI agent are reliant on simple rules that have nothing to do with the meaning of the sentence. For instance, the agent merely returns the first number it sees when the question uses the words how many.

3 Appendices- Enlarged Pictures



Question Word *Other conditions within the question*		Returned Object / Heuristic
who	->	Agent (and Coagent)
to exists	->	Reciever
from exists	->	Giver
how	->	Object-Adj
do is second word	->	Verb
many is second word	->	first number in sentence
second word is an ADJ	->	Object-Adj
second word is a distance noun	->	Distance
what	->	Object
second word is an ADJ	->	Object-Adj
where	->	Destination