Translating Indirect Responses to Direct Answers

Cynthia Chen (cchen16), Sherry Xu (sherryxu), Adis Ojeda (adiso)
6.863J/9.611J: Natural Language Processing - Professor Berwick

May 16 2018

Note: Reminder about our team's special situation consulted earlier with Professor Berwick.

Contents

1 Introduction

We are interested in studying conversations and the relation of sentences to each other, specifically in understanding how indirect responses answer questions. For example, in natural human dialogue, people may respond to questions in varied, complicated, and nuanced ways. Questions are a particularly important part of a conversation, and function to create discourse and an exchange of information. In this paper, we want to answer the following questions.

1. How do people understand indirect responses to questions?

2. How can a system be used to represent this behavior?

To accomplish this, we want to use a linguistics perspective to study conversations through an empirical view. We will use natural language processing concepts and semantic tools such as context free grammar parses and WordNet to translate indirect responses to direct responses. First, we parse the question and answer (QA) pair to identify interested components. Then we use WordNet and other techniques to compare each component to determine a final "Yes" or "No" response.

Though our final system only covers a narrow class of Yes-No QA pairs, we found some general rules that capture a good portion of QA pairs. When the interested components are easily identifiable and the intended meaning is literal, our system performs well. However, when there are context dependent co-references, our system is weaker.

2 Motivation

Real dialogue is often very different from formal written language. Oftentimes, humans answer questions by offering more information than the question requests, or in an indirect way that implies the true intended answer. Also, questions may be asked with different intended meanings (Athanasiadou). When formality is involved, people often ask and respond with more indirectness. We believe that observing Yes-No QA pairs found in typical human conversation provides the opportunity to reveal valuable insights about how we understand each other. Because we are interested in breaking down the process humans undergo to understand conversations, we will use a parser to transform the sentence into a tree structure. We will also use WordNet in the hope that direct answers can be found by looking at synsets and hypernym paths.

Advances in computational linguistics have helped in providing insight into thinking and intelligence. Language is the most essential and frequently used method of communication (Schubert). Translating indirect and seemingly disconnected responses into more direct and understandable language can also offer many benefits to machine translation, text summarizing, topic modeling and more. On the other hand, the knowledge gained from studying natural language can guide language systems in generating more natural, conversational sentences. Google has been studying how to enable machines to hold natural conversations with people for years. At Google I/O 2018, they announced their new artificial intelligence machine, Google Duplex, that can mimic question and answering in human conversations to conduct "real world" tasks over the phone such as reserving a table at a restaurant (Leviathan).

3 Literature Review

3.1 Linguistic Theory

In *Computational Models of Discourse*, the authors describe human discourse and discuss the work that has been done on understanding this complex behavior. Berwick defines discourse, generalized to communication, as exchanges between participants where they must reconstruct external messages to interior forms of the intended meanings. A message from the speaker to the recipient passes through stages of intent, interior form, utterance, parsing rules, literal meaning, and inference rules to finally reconstruct the speaker's intended message.

When a person asks a question, they have to carefully construct their question, as every utterance is a result of several speech acts. Then, when the other person responds to the question, they have to infer the intended meaning. For example, when a person asks about the truth of some statement and the statement happens to be false, the person is probably interested in the related true statement. The respondent is more likely to reply with the related true statement than simply the direct "No." Conversations like these require the speaker and recipient to share certain beliefs, knowledge, and preferences. This is a complicated process, and for our project, we intend to focus mainly on the logical rules required to reconstruct a message to its literal meaning.

3.2 Previous Works

A larger and more widely studied area of research is the field of Question Answering. The field of Question Answering focuses on building systems to automatically find answers in a large body of text documents to answer questions asked by humans. This is a very popular because it offers the potential of a more natural information retrieval system. It also has many other important implications and uses in computer systems.

In *Cogex: A Logic Prover for Question Answering*, Moldovan and his team use WordNet's hypernym and hyponym relationships to supplement a QA Parser. With the addition of other techniques they observed a 30.9 percent increase in the quantity of questions answered correctly. Using WordNet to supplement understanding and parsing of questions and answers is not a new idea, but we believe applying it to inspect the relationship between questions and natural indirect answers can offer meaningful insights. Since the addition of WordNet was successful to improving Cogex performance, we believe it will similarly benefit our question and indirect answer parsing.

Ulf Hermjakob describes in his paper, *Selectively Using Relations to Improve Precision in Question Answering*, how the addition of semantically parsed trees produces better results than a normal tree bank for a Question Answering system based on machine learning. Though the system he implements relies on machine learning and ours does not, we believe that the addition of semantically parsed trees is an effective supplement to our program.

Other papers, including *Learning Surface-Text Patterns for a Question Answering System*, and *Selectively Using Relations to Improve Precision in Question Answering*, outline potentially useful tools and techniques for better understanding of the relationship between questions and answers. Though they focus on Question Answering systems, we hope to integrate some of their insights into our indirect answer to direct answer project.

There has been previous research done of generating indirect answers to questions, which is the inverse of what we are attempting. In *Generating Indirect Answers to Yes-No Questions* by Green and Carberry, they generate indirect answers through two phases, planning the content of the answer and selecting the most concise answer. For planning the content of the answer, they break down indirect answers into multiple types such as clarification, substitution, and appeasement.

Finally, in *Indirect Responses to Loaded Questions*, Kaplan relied on a database of presumptions and knowledge about the questions. Using the database, he was able to query for connected subgraphs. Kaplan describes in *Computational Models of Discourse* a natural language database system, CO-OP, that answers questions cooperatively, as a human would in conversation. If the user makes an incorrect assumption in the question, the system will recognize this and correct the user. Instead of providing a nonsensical query result for an invalid question, the system will give an indirect answer that actually provides the user with meaningful information. This illustrates a certain class of answers that correct a statement found in the question, which is a feature of indirect answers we want to consider when designing our system.

4 Methodology and Implementation

4.1 Overview

Given an indirect answer to a Yes-No Question, our system attempts to translate the indirect answer into a direct "Yes" or "No" response. By implementing such a system, we hope to gain insight on the heuristics humans undergo to understand indirect responses in conversations.

For example, consider the following QA pair.

```
Q: Did the students in primary school finish their homework?
A: Only the students in secondary school completed their assignments.
```

As English speakers, we immediately understand that no primary students finished their homework. Because "finish" is synonymous with "complete" and likewise with "homework" and "assignments," we know the action in each sentence is the same. However, because the adjectives "primary" and "secondary" are not the same, we know the subjects performing the action are different. Therefore, the final direct answer is "No."

To create a system that models this behavior, we use Stanford CoreNLP to parse the QA pairs, and WordNet to find synsets, hypernyms, hyponyms, and antonyms. In conversations, people often use different words to represent the same idea – as done above with "homework" and "assignments" in the example above. These words are co-references, and can be understood with WordNet's synsets. This idea is extended to all parts of the sentence to construct the final direct answer.

4.2 Key Assumptions

For our system, the following assumptions are necessary:

1. The question answer pair can be correctly parsed by Stanford CoreNLP's constituency parser. Both sentences need to be correctly labeled and structured in order for our system to interpret them in the intended way.

2. The indirect answer to a question in some way implies the direct answer. In other words, there exists an intended "Yes" or "No" direct answer. We decided to make this crucial assumption because the distinction between indirect answer and non-related answer is often not clear. By assuming that the direct answers can be discerned from the indirect response, our system does not need to tackle the difficult task of determining when an indirect answer is too indirect.

   We feel like this assumption is reasonable because in typical human conversation, the reply to a question will be related and actually answer the question in some way. This is necessary for the conversation to be productive. While we provide answers to questions indirectly all the time, it is uncommon for humans to use an indirect answer to convey that they don't know the answer.

4.3 System Implementation

Our system can be broken down into two main components. First, we parse the QA pair to identify the interested parts of the sentence. Then we use WordNet and other techniques to compare each component between the question and answer sentence.

4.3.1 Parsing

To parse sentences, we decided to use Stanford CoreNLP tools. In addition to providing a constituency parser, which returns the hierarchical representation of the sentence, it also includes Tregex, a tool that is useful in identifying and matching patterns in the hierarchical representation of the sentence. Each component of the parse is also labelled with a Penn Treebank Tag to identify its part of speech. The following are parse trees for the example provided earlier.

```
Q: Did the students in primary school finish their homework?

    (ROOT
      (SQ (VBD Did)
        (NP
          (NP (DT the) (NNS students))
          (PP (IN in)
            (NP (JJ primary) (NN school))))
        (VP (VBP finish)
          (NP (PRP their) (NN homework)))
        (. ?)))
```

```
A: Only the students in secondary school completed their assignments.

    (ROOT
      (S (RB Only)
        (NP
          (NP (DT the) (NNS students))
          (PP (IN in)
            (NP (JJ secondary) (NN school))))
        (VP (VBD completed)
          (NP (PRP their) (NNS assignments)))
        (. .)))
```

In order to structure the QA pair for easy comparison, we extract the Noun Phrase (NP) and Verb Phrase (VP) from each sentence using Tregex. Because we are only handling Yes-No QA pairs, both the question and the answer sentence should have a complete NP and VP. By comparing the question's NP with the answer's NP and likewise with the VP's, we can look into whether the direct answer is "Yes" or "No." The following are the extracted NP's and VP's for the above QA pair example.

```
Q: Did the students in primary school finish their homework?

    Extracted Noun Phrase   (NP (DT the) (JJ primary) (NNS students))
    Extracted Verb Phrase   (VP (VBP finish) (NP (PRP their) (NN homework)))

A: Only the students in secondary school completed their assignments.

    Extracted Noun Phrase   (NP (RB Only) (DT the) (JJ secondary) (NNS students))
    Extracted Verb Phrase   (VP (VBD completed) (NP (PRP their) (NNS assignments)))
```

To format these tree-structure phrases into easily comparable structures, we transformed the tree into a dictionary where the dictionary keys are the Penn Treebank Tags. To preserve the hierarchical structure, only the direct children of NP or VP are added as keys.

Because there are multiple pluralities and types for nouns, to enable easy comparison in our system, any words with the label NN, NNS, NNP, NNPS, or PRP are inserted into the dictionary with the key "Noun." Similarly, because there are multiple tenses for verbs, any words with the label VB, VBD, VBG, VBN, VBP, or VBZ are inserted into the dictionary with the key "Verb." For any verb phrases that involve modals or are joined with a form of "to be," we only look at the base verb. For the verb phrase "have eaten," only "eaten" will be compared and stored in the dictionary. The following are the dictionary representations of the above QA pair.

```
Q: Did the students in primary school finish their homework?

    NP      {'DT': [Tree('DT', ['the'])],
            'Noun': [Tree('NNS', ['students'])],
            'JJ': [Tree('JJ', ['primary'])]}

    VP      {'NP': [Tree('NP', [Tree('PRP', ['their']), Tree('NN', ['homework'])])],
            'Verb': [Tree('VBP', ['finish'])]}

A: Only the students in secondary school completed their assignments.

    NP      {'DT': [Tree('DT', ['the'])],
            'Noun': [Tree('NNS', ['students'])],
            'JJ': [Tree('JJ', ['secondary'])],
            'RB': [Tree('RB', ['Only'])]}

    VP      {'NP': [Tree('NP', [Tree('PRP', ['their']), Tree('NNS', ['assignments'])])],
            'Verb': [Tree('VBD', ['completed'])]}
```

4.3.2 Comparing

Now that we have isolated the key components of interest in each sentence, we can use them to examine the relationship between the question and answer pair. Specifically, we determine if the question and answer pair match in the noun phrase and verb phrase, and in the elements the NP and VP are composed of.

Noun Phrase

To determine if the noun phrases in the QA pair match, our system first compares the main subject noun found during the parsing. If they match, it then considers potential modifiers to the noun subject such as determiners, prepositional phrases, and adjectives.

Verb Phrase

To determine if the verb phrase in the question and answer sentence match, our system first compares the main verb found during the parsing. If they match, it then considers potential elements that add specification to the verb action such as prepositional phrases, and adverbs, and direct objects.

For each type of component in the noun and verb phrase, we will examine its relationship between the question and answer.

Nouns and Verbs

Both nouns and verbs are essential to form sentences. In particular, all sentences our system handles must have a subject noun and an action verb. Additional nouns may be found in other locations, such as in prepositional phrases and as the direct object of the verb. We want to explore the relationship between corresponding noun and verb elements to determine if the answer statement is confirming or contradicting the statement asked in the question.

For example, if we examine the question and answer pair:

```
Q: Do some small penguins on the rock eat food?
A: Every little bird under the sun consumes food.
```

Our parser will determine the subject noun to be “penguin” in the question and “birds” in the answer. Next, to determine if the two noun subjects match, we use WordNet to see if there is a hypernym, hyponym, or same synset relationship between the question and answer noun. If any of these relationships exist, our system will understand it as evidence of possible co-referencing between QA pair. In the example, “bird” is an inherited hypernym of “penguin”, and in the answer, “birds” may be another way of talking about the “penguin” subject mentioned in the question. If the noun matches in this way, we move on to check that the modifiers of the noun – determiners, adjectives, and prepositional phrases – do not contradict each other between the question and the answer sentence. Our project has a similar process for comparing the verbs “eat” and “consumes” in the sentence above.

Determiners

The determiner of a noun can modify how the noun is referenced in the context of a question. Typical determiners such as “the” or “a” do not modify the subject in a way that is significant for our purposes. Other determiners do significantly affect the meaning of the noun, such as possessives, quantifiers, and indefinite. For our system, we decided to implement handling for the special determiners “every”, “some” and “no.”

Our system first examines the determiner modifying the noun in the question. This will help determine more specifically what the question is looking for. Depending on which determiner is modifying the noun in the question, we will check for different invalid determiners in the answer. Going off of the example earlier, the determiner modifying the noun “penguins” is “some.” Thus, as long as the determiner in the answer does not imply non-existence, in other words, it is not “no”, then the system considers it a match. In the example, ”every” does not provide evidence against the existence of ”some penguin”.

Prepositional Phrases

Prepositional phrases can be used in both the noun phrase and verb phrase to further specify their meaning. Since NP and VP can be modified by many different prepositional phrases, we limit our system to only be interested in those that have the same preposition in both the question and answer. For example, the prepositional phrases “on the rock” and “under the sun” add detail to the phrase in different ways. Only if the preposition is the same, for example “on” in “on the rock” and “on the stone”, do we want to check if the phrases refer to the same thing. Our system treats all comparisons of prepositional phrases in the question and answer, noun or verb, in this way, as long as they modify the same key part of the sentence. If neither the question nor answer has a prepositional phrase, or the same preposition as the other, the system does not invalidate the statement.

Adjectives and Adverbs

Adjectives and adverbs can modify the specific meaning of nouns and verbs, respectively. They can appear as just adjective and adverbs, or as a part of adjective phrases and adverb phrases. Similar to prepositional phrases, it is possible that different adjectives and adverbs can modify the same noun or verb. For example, the phrase "cute penguin" and "short penguin" do not necessarily refer to different penguins. In addition, adjectives and adverbs are not categorized easily into hypernym and hyponym relationships like nouns and verbs. Thus, for an adjective or adverb in the answer to disagree with that of the question, it would need to be a direct antonym. Finally, if an adjective or an adverb does not exist for an element in the question or answer, the adjective and adverb does not need to be checked.

Negation

Our system supports negation for verb and adjective phrases. An example of negation for the VP and adjective phrase is:

```
VP negation:       The bird doesn't sit on the rock.
ADJP negation:     The penguin is not very quick.
```

To handle negation in adjectives and verbs, the system checks for the existence of negative words that modify the interested element. If negation exists in neither the question and answer, or both the question and answer, they cancel out, and do not have an effect if the negated elements match. If negation exists in the corresponding adjective of the question or the answer sentence, the other adjective must either also contain negation, or have an antonym relationship. Negation handling in the verb is similar, but in the case of only one of the question or answer having negation, an antonym relationship is not necessary to maintain the match. The verbs only need to be different. Our system cannot yet handle double negation.

4.3.3 Determining the Direct Answer

Using the components discussed above, our system can determine the direct answer for the following example:

```
Q: Do some small penguins on the rock eat food?
A: Every little bird under the sun consumes food.
```

Our system first parses the sentence and extracts the noun phrase and verb phrase. Using these phrases, it then condenses them into a dictionary of key components. Next, it compares the noun phrase and verb phrase separately.

```
Comparing Noun Phrases
Q NP:  (NP  (NP (DT some) (JJ small) (NNS penguins)) (PP (IN on) (NP (DT the) (NN rock))))
A NP:  (NP (NP (DT Every) (JJ little) (NN bird)) (PP (IN under) (NP (DT the) (NN sun))))
-NOUN-      Q: 'penguins'              A: 'bird'              <PASS (hyponym)>
-DET--      Q: 'some'                  A: 'Every'             <PASS>
-PP---      Q: 'on the rock'           A: 'under the sun'     <PASS (different prep)>
-ADJ---     Q: 'small'                 A: 'little'            <PASS (same synset)>
NOUN SIMILARITY True
```

```
Comparing Verb Phrases
Q VP:  (VP (VB eat)  (NP (NN food)))
A VP:  (VP (VBZ consumes) (NP (NN food)))
-VERB-     Q: 'eat'                 A: 'consumes'         <PASS (same synset)>
-NEG--     Q: ' '                   A: ' '                <PASS (no neg)>
-DO---     Q: 'food'                A: 'food'             <PASS (same synset) >
VERB SIMILARITY True

Direct Answer: Yes.
```

After the relationship between the QA noun phrases and QA verb phrases are determined, our system then checks if both the NPs and VPs have similarity. If they both match, then the direct answer is “Yes”. Otherwise, the direct answer is “No”.

4.4 Challenges

Our initial system involved the use of dependency parsing. Unlike constituency parsing, dependency parsing immediately identifies the subject and the verb. There is no need to go through multiple layers of the NP tree or VP tree to find the main subject or action word. The dependency parse also labels negation words. Although this labeling method is more convenient, when we started exploring how adjectives, adverbs, and determiners interacted with the main subject and action, we realized the dependency referencing system was inefficient. Every time we encountered a new tag in the question, we needed to search the entire structure to find the potential matching tag in the answer. Dependency parsing also loses c-command relation within sentences. Therefore we migrated our work to a constituency parser whose hierarchical structure allows for a much more modular basis for our work.

Another challenge we encountered was finding the appropriate method to compare different parts of speech using WordNet. At first we compared everything by analyzing if the two words were hypernyms, hyponyms, or in the same synset. This method was sufficient for comparing nouns and verbs. However, for adjectives, adverbs, prepositions, and determiners, this method did not accurately reflect how humans understand relationships. For adjectives describing nouns, a “small cat” can also be an “orange cat.” We therefore switched to comparing adjectives using antonyms. Similar methods were implemented for other parts of speech.

The third main challenge we encountered were cases when we compare components in the QA pair that are labeled differently. The components can be located in different levels of the tree-structure of the sentence. Some of these cases were solved by inserting the component into the dictionary using a more general key such as “Noun” and “Verb.” Other cases, such as adjective phrases and adjectives are more nuanced because the adverb in the adjective phrase can potentially affect the comparison. We approached this problem by making our system more modular and allowing cases of comparison when either the question or answer’s component did not exist.

4.5 Possible Additions

Some additions to our system that we planned but did not get to implement due to our team's situation include pronoun integration with WordNet and handling for clauses and conjunctions. These additions would widen the scope of questions our system can understand, and improve the performance.

Pronoun integration with Wordnet

To implement this, we would construct our own pronoun hierarchy, and use it in a similar way to WordNet's hypernym homonym and synset relationships. Similar to Noun determiners, we would need to also include logic to handle indefinite pronouns such as "everyone." This pronoun hierarchy would also be useful in examining possessive determiners such as "his" and "theirs."

Clause

Clauses appear frequently in human conversation and are important to provide more detail about the context. Since our system can only use the sentences provided, examining clauses will be important to understand context. Our system would handle clauses similar to how it understands prepositional phrases. Clauses are similar to prepositional phrases in that they add specificity to the element they are attached to. For example, in the following sentences, the prepositional phrase and the clause both describe the dog.

```
PP:       The dog on the field runs fast.
Clause:   The dog that runs on the field is fast.
```

Conjunctions

Oftentimes, we ask about multiple ideas in one question. These ideas are usually closely related and connected with a conjunction such as "and" or "but". If the sentence contains a conjunction, we will break down the sentence into separate statements. As long as no statement of the answer contradicts any statement in the question, and the indirect answer supports at least one statement in question, then our system will consider the direct answer to be "Yes".

5 Testing

5.1 Testing Procedure

To test our system, we created our own corpus of 35 QA pairs. For each test QA pair, we first manually labeled the expected direct response as "Yes" or "No." Then, we compared the expected response to the actual response from the system.

Because there is no readily available external data set of indirect answers to questions, we decided creating our own corpus would be the best protocol even though our test set may be biased. To develop our test set, we each wrote statements about randomly generated nouns and then used those statements to reverse generate questions. Therefore, each QA pair was disjointedly written by two people, hopefully eliminating some bias and overfitting problems. We decided to create a test set of 35 QA pairs to limit the manual labor involved in writing sentences, determining the expected result, and investigating the parse.

Even though our test set may not fully generalize to all types of Yes-No QA pairs, it gives valuable insight into our system's strengths and weaknesses. Of the 35 QA pairs we created, 19 performed as expected, 10 produced the wrong answer, and six produced errors due to CoreNLP parsing. Our system performs better than a random Yes-No answer generator, but encounters errors more frequently.

5.2 QA Pairs: Correct Response

We have 19 out of 35 QA pairs that return the expected result. However only 14 of the 19 correct responses are due to proper comparison in our system. Most of these 14 QA pairs perform as expected in our system due to accurate matching between sentence components and direct WordNet relationships. For example:

```
Q: Is it true that punctuation is used in many essays?
A: The semicolon is used a lot in papers.
```

Our system correctly identifies that "punctuation" is a hypernym of "semicolon," therefore "semicolon" and "punctuation" are co-references. In addition, "paper" is a hypernym of "essay," so "paper" and "essay" are also co-references. Because the subject, action, and direct object all match, the direct answer is "Yes."

Our system also correctly handles some cases of negation and quantifiers such as:

```
Q: Do all elephants run for president?
A: Male elephants do not run for president.
```

First, because the quantifier "all" includes any type of elephant, we know the noun phrases match. Next, because the verb phrase in the answer sentence is negated, we immediately know the verb phrases do not match. The final direct answer is "No" because the verb phrases do not match.

The other 5 QA pairs compare unrelated components to each other. This problem will be discussed in the next section. For these 5 cases, however, the correct response was returned because of interactions with other components.

5.3 QA Pairs: Incorrect Response

For the 10 QA pairs that produced the wrong answer, they either compare the wrong components, are difficult to characterize the relationships using WordNet, or include components our system can not handle.

Improper Comparisons

```
Q: Does the author of the book write many books for kids?
A: The author of the book is very famous for writing children books.
```

Intuitively as an English speaker, the direct response is "Yes." Even though the indirect answer's verb is not a form of "write," because "writing children books" is inside the prepositional phrase, we know the author writes books. Our system does not know to compare prepositional phrases and verbs. It only compares similar components.

WordNet Limitations

```
Q: Is it true that the ladys clothing is very beautiful?
A: The girls dress is really pretty.
```

The expected answer for this example is “Yes,” because we can imagine the lady and the girl being the same person. However, in WordNet, neither are in each other’s hypernym path. They do, however, both share the hypernym “woman.” It is hard to make this a rule for all words because it not always true that two words can be co-references if they share a hypernym.

Unimplemented components

```
Q. Did the coach tell your grandmother to exercise?
A. A coach told my granny to visit the new room.
```

The main difference between the question and answer is “to exercise” and “to visit.” The expected answer is “No,” but because our system does not handle infinitives, these words are skipped over in the comparison and the system returns “Yes.”

5.4 QA Pairs: Parse Error

Lastly, the third result from our test set was parse errors. Six QA pairs returned parse errors because CoreNLP incorrectly labeled sentence components. Many verbs in sentences were mislabeled as nouns. Because our system relies on extracting the VP from the sentence, it cannot proceed without an identifiable VP in the consistency parse.

```
Sentence    Does every animal roll?
Expected Parse
            (ROOT
              (SQ (VBZ Does)
                (NP (DT every) (NN animal))
                (VP (VB roll))
                (. ?)))
Actual Parse
            (ROOT
              (SQ (VBZ Does)
                (NP (DT every) (NN animal) (NN roll))
                (. ?)))
```

6 Evaluation

Based on our testing, our system works slightly better than a random Yes-No answer generator. Our system seems to be accurate for cases when the parser clearly identifies the main subject and action in the NP and VP. It also deals with negation, antonyms, and logic cases. However, it is still limited in its ability and far from close to replicating human ability. There are too many classes of indirect responses which make encoding a generalized set of rules to represent all cases difficult.

Our testing is based on our corpus of self-generated 35 QA pairs because there is no readily available external corpus of indirect QA pairs. The base line for testing was our manually labeled "Yes" or "No." Although the result of our testing may be biased and not representative of its accuracy, it provides insights into classes of indirect questions that our system does not cover.

Our system requires full sentences with both NP and VP and thus is lacking when either a component is not provided in the indirect answer or the required context is not included in the answer or WordNet. When an interested component is not provided in the expected location such as "write" and "writing" in the following example:

```
Q: Does the author of the book write many books for kids?
A: The author of the book is very famous for writing children books.
```

our system gets confused. It does not understand that we can compare verbs and prepositional phrases. We cannot, however, encode this rule for all QA pairs. As English speakers, we intuitively understand what components are comparable and incomparable. Also, in face-to-face conversation, humans can better understand intentions behind questions through tones and domain context.

Most of the comparisons our system makes are based on WordNet because WordNet is one of the best, currently available lexical database that represents relations between words. While implementing our system, we had to create extensions to complement WordNet to represent cases with negation and quantifiers. WordNet also does not include pronouns. Although pronouns do not have synsets, there is a pronoun hierarchy. For example, something that applies to "everyone," also applies to "she." This not apply the other direction though. Pronoun integration, as mentioned in possible additions section, was not a feature we had time to implement. Our current system considers two words to be co-references if either of the words appear in any of the other's hypernym paths. However, as a human, it is usually up to their own judgment to determine how far down the hypernym path a word can be to still be considered a co-reference. Also, humans immediately understand which of the word's many synsets is actually used in the sentence. Our system looks at all possible synsets, thus leading to inaccuracy and improper generalization. When comparing "red" and "blue" in terms of the color of an object, it is possible the object can be both "red" and "blue" such as the American flag. However, when comparing "red" and "blue" in terms of political parties, they are antonyms. WordNet cannot distinguish differences regarding domain context.

Our system currently does not extensively consider context. Existing work that has been done to encode context include creating a question answering database such as in START or the system in 6.863J Lab 3. These databases include event structures and remember components such as the subject, action, and beneficiary of the sentences. They can encode more information, but the hierarchical structure of sentences may be harder to maintain in the database. When comparing our system to START, START seems to require more information than our system does before answering "yes" or "no." START commonly replies back with "I don't know." We did not compare our system to the 6.863J Lab 3 system because Lab 3 has a limited vocabulary.

7 Conclusion

Our main takeaway from this project is that understanding and determining the direct responses from indirect answers is complex and very nuanced. There are general rules to capture a large portion of literal behavior, but it is much harder to represent the interactions of different rules, to know how context affects the result, and to interpret meaning and intentions of indirect answers.

Overall, we learned that humans converse in wide ranging complex ways. Even when limiting our scope to understanding indirect answers to questions, we find that indirect responses can convey the same direct answer through a variety of methods.

In our system, we specifically examine co-referencing through hypernyms, hyponyms and same synset relationships, logical negations, quantifiers, prepositions, adjectives, and adverbial modifiers. Despite our efforts, we do not believe our system exactly replicates the way in which humans interpret indirect answers. In addition, when the answer is too indirect, even humans will disagree on the correct direct implication of the response. Unlike our system, humans don't interpret an indirect response by breaking it down into all the elements and handling them separately. More accurately, humans interpret the response in real time, as they receive it, until something convinces them of the correct direct answer. Even though our rule-based approach works, if we were to expand further on this, our system cannot account for all of the possible ways to answer a question indirectly. Especially since an indirect answer can be almost anything besides a direct "Yes" or "No." Finally, our system, like many others, depend on a parser that can parse the sentences correctly. Until a parsing system is built that can determine the correct hierarchical parse of a sentence without fail, programs like ours that are reliant on having the correct parse will not work correctly as intended all the time.

If we had more resources to invest into this project, there are many future considerations we would add. Firstly, we would want to extend our system to handle more types of question answer pairs, such as wh-questions, and find better generalizations between different types of questions and answers. Another addition we think would be useful is the implementation of a database or memory structure to remember context. This would mean our system would take in multiple sentences as input, and use the information from previous inputs to gain a better understanding of the entire context behind the responses. We also are interested in investigating WordNet and adding our own extensions to better represent interested relationships between words. For example, all color adjectives fall in the same category of having color, yet using WordNet to understand the relationship between the different color adjectives is diffficult. Lastly, we would investigate degrees of indirectness, and determine a threshold for how indirect an answer can be.

8 References

Athanasiadou, Angeliki. "The Discourse Function of Question" *Pragmatics. Quarterly Publication of the International Pragmatics Association (IPrA)* 1.1, 1991: 107-122. https://journals.linguisticsociety.org/elanguage/pragmatics/article/download/332/332-618-1-PB.pdf

Brady, Michael and Berwick, Robert C. (1984): *Computational Models of Discourse.* Second printing. The MIT Press.

Green, Nancy and Carberry, Sandra. "Generating Indirect Answers to Yes-No Questions" *The Seventh International Workshop on Natural Language Generation.* Association for Computational Linguistics, 1994. http://www.anthology.aclweb.org/W/W94/W94-0322.pdf

Hermjakob, Ulf. "Parsing and Question Classification for Question Answering." *Workshop on Open-domain question answering-Volume 12.* Association for Computational Linguistics, 2001. https://pdfs.semanticscholar.org/80e9/caf96d3730915bb3ff48772bc2689dfce62d.pdf

Katz, Boris and Lin, Jimmy. "Selectively Using Relations to Improve Precision in Question Answering." *EACL 2003 Workshop on NLP Processing for Question Answering* April 2003. Budapest, Hungary https://groups.csail.mit.edu/infolab/publications/Katz-Lin-EACL03.pdf

Leviathan, Yaniv, and Yossi Matias. Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone. *Google AI Blog*, 8 May 2018, ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html.

Moldovan, Dan; Clark, Christine and Harabagiu, Sanda. "COGEX: A Logic Prover for Question Answering." *HLT-NAACL 2003 Main Papers*, pp. 87-93 Edmonton, May-June 2003. https://pdfs.semanticscholar.org/1428/7f83e7bbfcc2de9c14be57fc39ef5026cb89.pdf

Ravichandran, Deepak and Hovy, Eduard. "Learning Surface-Text Patterns for a Question Answering System." *40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, July 2002, pp. 41-47 http://www.aclweb.org/anthology/P02-1006?CFID=26171221&CFTOKEN=5cb04de659ec238c-30CE109C-05E3-7CEF-D8188B998112EE18

S. Jerrold Kaplan Kaplan, S. Jerrold. "Indirect Responses to Loaded Questions" *The 1978 workshop on Theoretical issues in NLP.* Association for Computational Linguistics, 1978. https://aclanthology.info/pdf/T/T78/T78-1029.pdf

Schubert, Lenhart, "Computational Linguistics", *The Stanford Encyclopedia of Philosophy (Spring 2015 Edition)*, Edward N. Zalta (ed.), https://plato.stanford.edu/archives/spr2015/entries/computational-linguistics.