Contents

1 Introduction

Well-formed constructions of a natural language have both an observable representation, written or spoken words, and a semantic representation, the meaning behind these utterances. Taken together, the observable and semantic represenations of a phrase constitue a form-meaning pair. Knowledge of that language is then the ability to translate between form and meaning, interpreting the observable form of an construction and producing new observable representations in kind from semantic representations. While to us the connection between a form and its meaning is intuitive, to machines this relationship is not innate and must then be taught.

One challenge in teaching machines to understand language comes from words or phrases whose meaning is necessarily context-dependent. Simple words, like proper nouns or verbs, may have meanings which are derivable from the words themselves. For instance, consider a simple language consiting of names and intransitive verbs like the one shown below in (1).

(1) a. [[Alice]] = the girl named Alice
b. [[Bob]] = the boy named Bob
c. [[thinks]] = the act of thinking

The meanings of sentences in this simple language are nicely composed of the meanings of the constituent words. For example, in (2) we see that the sentence "Alice thinks" is understood of one knows the meanings of "Alice," "thinks," and the rules about how sentences are formed from constituent parts.

(2) [[Alice thinks]] $\approx$ [[Alice]] + [[thinks]] = the girl named Alice thinks

Not all words, however, have this nice property that their meanings are independent of the words surrounding them. Consider the case of anaphora, where reflexive pronouns refer to other nouns in a sentence, like in (3).

(3) a. Alice sees herself
b. Claire sees herself

The sentences here can't be easily interpreted from the meanings of the individual words in the same way that (2) can. For one, it's clear that the word "herself" doesn't have a meaning independent of "Alice" or "Claire." Furthermore, it's apparent that the meaning of "herself" changes depending on the context. These are in fact the defining properties of reflexive pronouns: they must be bound by some independently-defined noun in order to have meaning. Cases like this pose challenges for machines being trained to translate forms into meaning.

Another problem facing linguists and computer scientists is trying to figure out what exactly is happening when machines "learn" a task. Defined extentially, one might think that a machine has learned a task when it can successfully complete the problems set before it. That is, if we give a machine a list of sentences and ask it to translate those sentences into semantic representations, and it does so correctly, we might think that the machine has learned how to interpret language.

2 Background

2.1 Form-Meaning Pairs

2.2 Anaphora and Binding Theory

2.3 Neural Networks and Anaphora

The question of whether or not neural networks can learn the semantic representations of anaphora is not new. [1] posed this problem and investigated whether or not a real-time RNN network architecture could predict the meanings of anaphors. Their results found that real-time networks were unable to successfully.

3 The Seq2Seq Network Architecture

3.1 Specifying the Form and Meaning domains

In order to generate training data consisting of form-meaning pairs, I use a Featural Context-Free Grammar to proceduraly generate sentences and then parse them into semantic representations using the `nltk` Python library. The Form domain is then all possible sentences generated by the grammar, while the meaning domain is the corresponding collection of semantic representations.

An example, minimal grammar is given below along with the sentences is generates and their associated semantic representations.

```
# file: grammar.fcfg
% start S

# Grammatical Rules
S[SEM = <?pred(?subj)>] -> NP[SEM = ?subj] VP[SEM = ?pred]
VP[SEM = <?v(?obj)>] -> V[SEM = ?v] NP[SEM = ?obj]

# Lexical Rules
NP[SEM = <\P.P(alice)>] -> Alice
VP[SEM = <\x.know(x)>] -> knows
```

4 Experiment: Generalizing Anaphora to new Lexical Items

One of the simplest types of sentences involving reflexive pronouns are sentences containg only names, transitive verbs, and reflexive pronouns. An example of one such sentence is shown below in (4).

(4) Alice sees herself.

We define a simple predicate logic where verbs are mapped to predicates, and subjects and objects are mapped to the arguments to those predicates, as in (5) below.

(5) Bob sees Alice → see(bob, alice)

In cases where the object of the verb is a reflexive pronoun, like in (4), the subject and object arguments to the predicate are simply the subject of the sentence, as in (6).

(6) Alice sees herself → see(alice, alice)

Intransitive sentences are mapped to predicates with a single argument, as in (7).

(7) Bob sleeps → sleep(bob)

In order to test whether ot not a Seq2Seq model can be trained to generalize knowledge of reflexive sentences, we will selectively withhold certain sentences generated by the language grammar from the training set and see how the network performs on these novel cases. In this set of experiments, we will examine whether the Seq2Seq models can learn to parse 'Alice-reflexive' sentences like in 4, which are of the form of (8) below.

(8) Alice *verbs* herself

Our Seq2Seq model is made of recurrent encoders and decoders and can optionally implement attention. We conduct these tests using SRN, GRU, and LSTM architectures with No Attention, Additive Attention, and Multiplicative attention to gauge the effect that network architecture has on our models' abilities to learn to interpret reflexive pronouns and generalize this knowledge to new cases. For each combination of recurrent unit and attention, we train three models separately to see if there is any variance in the networks' abilities.

Grammar		
S	→	Name VP
VP	→	V_i | V_t Name | V_t Refl
Name	→	Alice | Bob | … | Zelda
Refl	→	himself | herself
V_i	→	walks | sleeps | eats | runs | sings | dances | flies | slumbers
V_t	→	sees | meets | likes | dislikes | throws | notices | knows

Table 4.1 Context-free grammar used for Alice-* experiments.

The sentences produced by this grammar are matched to semantic representations in the following way.

x V_i	→	*verb*(x)	Alice sleeps	→	sleep(alice)
x V_t y	→	*verb*(x, y)	Bob knows Claire	→	know(bob, claire)
x V_t Refl	→	*verb*(x, x)	Zelda sees herself	→	see(zelda, zelda)

Table 4.2 Semantic representations of generated sentences for Alice-* experiments.

For example, examples in (9) show how various sentences produced by the grammar are parsed into form-meaning pairs.

(9) a. Alice sleeps → sleep(alice)
b. Bob sees Claire → see(bob, claire)
c. Delilah knows herself → know(delilah, delilah)

4.1 Alice-α: Can Alice know herself?

The Alice-α experiment tests whether or not a Seq2Seq model can be trained to generalize knowledge of simple sentences containing transitive verbs and reflexive pronouns to novel subject-reflexive sentences. Alice-reflexive sentences like those in 4 are excluded from the training dataset. All other sentence types generated by the grammar (including intransitive sentences with and without Alice, transititive sentences with Alice in subject position, and transitive sentences with Alice in the object position) are included in the training set.

Alice *verbs* herself

Table 4.3 Sentences withheld from Alice-α training set.

4.2 Alice-β: But wait, doesn't Alice know Alice?

In the Alice-α experiment, we witheld all sentences of the form "Alice *verbs* herself" from the training data in order to see if the network could successfully generalize knowledge of other reflexive sentences to a new subject. While successful at this task, it must be noted that although the network had never encounted a reflexive sentence whose subject was "Alice" in training, it did encounter sentences whose semantic representation is identical to that of an Alice-reflexive sentence. Namely, sentences of the form "Alice *verbs* Alice" were present in the training set, and both these and Alice-reflexive sentences yield semantic represenations of verb(alice, alice), as shown below in (10).

(10) a. Alice knows Alice → know(alice, alice)
b. Alice knows herself → know(alice, alice)

Although the network was never encountered Alice-reflexive sentences like those in (10a) during training in the Alice-α experiment it did encounter Alice-Alice sentences like (10b). Since both types of sentences have the same semantic representation, the presence of Alice-Alice sentences may bais the network in favor of producing the correct semantic representation since the network can learn that representations of the correct form exist.

In order to examine whether the presence of Alice-Alice sentences impact the networks' abilities to learn and generalize knowledge of anaphora, we conduct a second experiment, Alice-β. In Alice-β, we withold both Alice-herself and Alice-Alice sentences from the training data. Withholding both types of sentences from the training data forces the network to generate an entirely novel output; whereas the networks in Alice-α had to generalize to a new input, networks in Alice-β must generalize to new inputs and new outputs.

4.3 Alice-γ: What if nobody knows Alice?

Since the networks perform well at the task of generalizing knowledge of anaphoric representation to a new subject, it is natural to ask just how impoverish the network's stimulus may be while still being able to correctly interpret the sentences of our language. To this end, we conduct another experiment, Alice-γ, in which we exclude all sentences with Alice

in the object position, like (11), from the training data along with the Alice-reflexive and Alice-Alice sentences withheld in Experiment Alice-β.

(11) Bob knows Alice → know(bob, alice)

Alice *verbs* herself	Alice *verbs* Alice	*Person verbs* Alice

Table 4.4 Sentences withheld from Alice-γ training set.

4.4 Alice-δ: What if Alice doesn't know anyone?

4.5 Alice-ε: Who's Alice and who's Claire?

5 References

[1] R. Frank, D. Mathis, and W. Badecker. *The Acquisition of Anaphora by Simple Recurrent Networks*. 18 June 2013. *Language Acquisition*.