Lexical Semantics Analysis of WordNet Through Humor Recognition

Zoe (Siyu) Lu (siyulu@mit.edu), Mahi Shafiullah (nshafiul@mit.edu), Katie Sedlar (ksedlar@mit.edu)

***Abstract*— Over 30 years ago, a group of psychologists and linguists at Princeton developed WordNet, a lexical reference system that aimed to categorize lexical information in terms of meaning, rather than form. We seek to analyze the lexical semantics of WordNet by viewing it through the lens of computational humor. To this end, we present a program that analyzes the "Walk into a bar" jokes by using WordNet to identify the semantic relationships between the key words/phrases in these jokes. The performance of WordNet in humor understanding is compared to that of ConceptNet as well as to humans. While analyzers based on both databases had some success recognizing puns, ConceptNet is better suited for the task due to its much richer word-relation representation. Nevertheless, both analyzers had high rate of false positives and neither database allows us to construct the situation that a given joke depicts. These results point to potential improvements not only in our programs, but also in lexical databases such as WordNet. Specifically, WordNet would better portray human psycholinguistics if it could manage to capture the context of a given text with lexical information on individual words. For the purpose of processing practical, context-oriented text, it would benefit from an extension that includes informal, commonsense knowledge as provided in ConceptNet.**

I. Introduction

An on-line lexical reference system whose development started in 1985, WordNet distinguishes itself from a traditional dictionary or thesaurus in that its underlying database aims to represent lexical memory instead of lexical knowledge. In this way, it claims to better simulate how humans retain lexical information. Specifically, it stores words based on their semantic relations with each other, thus providing its own unique description of lexical semantics.

Lexical semantics begins with the mapping between word forms (the physical utterance or inscription of words) and word meanings (the lexicalized concept expressed by the form) [1]. In human languages, these mappings are many:many. Multiple words forms that express the same meaning are synonymous, whereas a word form that expresses multiple meanings is polysemous. Representation of forms can be achieved by inscription, but representation of meanings is much trickier. WordNet does the latter through semantic relations, and the principles governing these relations differ for words with different parts-of-speech. These principles are detailed in the Organization of Nouns, Adjectives, and Verbs in WordNet under Section III.

In our project, we analyzed and evaluated how effective WordNet's approach to lexical semantics reflects the psycholinguistics of human. We do so via the lens of humor recognition, as it is widely acknowledged as a hard problem in terms of semantic interpretation, which provides a good starting point for us to identify the existing strengths and weaknesses of WordNet.

In particular, we focused on jokes that follow the "Walks into a bar" format (e.g. *A tennis ball walks into a bar. The barman says, "Have you been served?"*) and used WordNet to determine the semantic relations between key words that render the text humorous. We then compared WordNet's performance with that of ConceptNet as well as humans to evaluate the system's effectiveness in representing lexical semantics based on human psycholinguistics. Finally, based on our evaluation on how well WordNet mimics what humans do, we propose suggestions for how this tool can be improved.

II. Motivation

Computational humor is a relatively new field that has been regarded as one of the final frontiers of artificial intelligence, as humor requires self-awareness, spontaneity, linguistic sophistication, and empathy, which are lacking in robots and computational models [2]. Motivations for computational humor have been discussed in many previous research, including considerations such as making the computer user-friendlier and more persuasive, enhancing human-computer interaction, developing better intelligent agents, improving second language learning systems, facilitating electronic advertising and so on [3].

Research in computational humor can be divided into humor generation and humor recognition. We are particularly interested in the latter because it provides us with a unique lens to evaluate systems that have been widely employed in natural language processing tasks, such as WordNet. Specifically, results from our project will be useful at least for the following two purposes.

First, as described in the Semantics of Humor under Section III, humor understanding poses additional challenges than normal text understanding, and thus complete humor understanding seems to be human-specific as of now. Since WordNet aims to be a database that reflects the psycholinguistics of human, evaluating its performance on humor recognition will point directly to where it is still lacking in its lexical semantics as compared to humans. Such evaluation will help us improve lexical reference tools such as WordNet, so they can get closer in their goals of better approximating humans in understanding speech and humor in particular.

Second, WordNet and other similar tools are currently widely employed in applications that motivate computational humor research (e.g. human-computer interaction). To render these applications more effective, we need to ensure the underlying lexical database is best suited for them. Therefore, motivations for this project are complementary in the sense that humor recognition allows us to pinpoint the strengths and weaknesses of WordNet while improvements of WordNet are crucial to refine existing applications that involve humor understanding.

In evaluating WordNet, we rely on the Semantic Script Theory of Humor, as detailed in the Semantics of Humor under Section III. We seek to determine whether lexical semantic analysis by WordNet successfully identifies the key humorous text under this theory. We also benchmark WordNet's performance against that of ConceptNet to see if more complex representation of ontology can improve humor recognition. Finally, we discuss how these systems' performance compare to what humans are able to do and offer suggestions for improvements, as prompted by our two main motivations.

III. Previous Work

Organization of Nouns, Adjectives, and Verbs in WordNet

Five seminal papers—namely, "Introduction to WordNet: An On-line Lexical Databse" [1], "Nouns in WordNet: A Lexical Inheritance System" [4], "Adjectives in WordNet" [5], "English Verbs as a Semantic Net" [6], and "Design and Implementation of the WordNet Lexical Database and Searching Software" [7]—lay the foundation for our work. To analyze and evaluate WordNet's representation of lexical semantics, one first has to understand the principles behind semantic relations WordNet employs. These principles differ for words with different parts-of-speech. Since our project focuses on nouns, adjectives, and verbs, this section presents the fundamental semantic relations underlying these three categories.

As the second paper suggests, lexical inheritance is the core principle for semantic relations between nouns. Such inheritance structure relies on hyponymy, relating a word with a more specific meaning to its superordinate. For instance, {*maple*} is a hyponym of {*tree*}, which in turn is a hyponym of {*plant*}.

Regarding semantic components, WordNet partitions the nouns with a set of twenty-five semantic primes (Miller, 1993). Distinguishing features are added to the inheritance hierarchy to supply details for differentiating one concept from another. Miller outlines three distinguishing features: attributes, parts, and functions. For the example word *pen*, its attributes may be *lightweight, black*, parts *ink, outer case*, and function *write*. As semantic relations are implemented as pointers in WordNet, attributes require pointers from nouns to adjectives, parts require pointers from nouns to nouns, and functions require pointers from nouns to verbs [4]. Attributes are described in the following paragraph; functions are currently not supported; this paragraph concerns primarily with the part-whole relation, meronymy.

While there are several types of meronyms, three types are coded in WordNet: A is a component part of B; A is a member of B, and A is the stuff that B is made from [4]. In addition to meronymy, antonymy (semantic opposition) is also represented in WordNet, though it is not a fundamental organizing relation. The three semantic relations—hyponymy, meronymy, and antonymy—link nouns into a highly connected network. Aside from semantic relations, WordNet also includes two key lexical relations: synonymy and antonymy. The grouping of synonymous words is called a synset in WordNet.

Bipolar oppositions (antonymy) serve to organize adjectives [5]. WordNet divides adjectives into descriptive and relational. The former are essentially values of attributes for nouns, as mentioned above. WordNet contains pointers between descriptive adjectives and the noun synsets that refer to the appropriate attributes. The authors believe nearly all attributes are bipolar. Under this formulation, all descriptive adjectives either have direct antonyms or indirect antonyms (by being synonyms of adjectives that have direct antonyms). Relational adjectives—adjectives that relate/pertain to or are associated with some noun—are semantically different from descriptive adjectives because they do not relate to an attribute [5]. WordNet maintains a separate file of them with pointers to the corresponding nouns [5]. Color adjectives are treated as a special case and are organized by dimensions of color perception: lightness, hue, and saturation [5].

Lexical entailment governs the organization of verbs [6]. WordNet divides verbs into fifteen files, with fourteen of them being semantic domains. Lexical entailment holds between V1 and V2 when the sentence *Someone V1* entails the sentence *Someone V2*, so there is no conceivable state of affairs that could make *Someone V1* true and *Someone V2* false [6]. For instance, *snore* lexically entails *sleep* because *He is snoring* lexically entails *He is sleeping*.

WordNet utilizes four types of entailment: (1) +troponymy (co-extensiveness) (*To V1 is to V2 in some particular manner*), (2) -troponymy (proper inclusion), (3) backward presupposition, and (4) cause [6]. Rather than explaining the technical differences among these four kinds of entailment (which can be found in the original paper), we simply list examples for each type and discuss them in details as necessary in the discussion: (1) *limp-walk*, (2) *snore-sleep*, (3) *succeed-try*, and (4) *give-have* [6].

To capture the correlations between a verb's semantic make-up and its syntax, WordNet includes one or several sentences frames for each verb synset to indicate what kinds of sentences these verbs can occur in [6]. The author argues the semantics of the troponyms can also illustrate the syntactic distinction between the alternations of a verb. For instance, the intransitive troponyms of *eat* all convey the sense *eat a meal* while the transitive troponyms of *eat* all convey the sense *ingest in some manner*.

Leveraging the power of WordNet, much work is done to advance humor recognition [1][8][9]. However, to our knowledge, previous research focuses solely on using WordNet to design and improve automatic humor recognition programs. We present the first attempt to analyze semantic relations among words in humorous expressions with WordNet to illuminate the strengths and weaknesses in WordNet's approach to lexical semantics. We also intend to offer suggestions for how to improve WordNet's current design so as to better align with how human understand humor.

Semantics of Humor

The most prevalent theory behind humor understanding is Victor Raskins 1985 Semantic Script Theory of Humor (SSTH) and its later extension in 1995, the General Theory of Verbal Humor (GTVH).

Humor understanding calls for semantic interpretation of sentences rather than individual words. The semantic interpretation of a sentence is calculated on the set of one or more combinations of scripts compatible with all the lexical items making up the sentence [10]. A script is a typical sequence of events that occur in a standard situation.

Based on SSTH, a piece of text is humorous if it is compatible with at least two different and in a sense opposite scripts for the scope of the text [10]. Consider the example: *A man walks into a bar. Ouch.* There are two key associated scripts activated by the word *bar*. The first describes entering a bar, whereas the second one describes hitting a pillar-like object [3]. The first script elicits lightheartedness or pleasure, yet the second script elicits pain, thus establishing the oppositeness of the two scripts [3].

The GTVH introduces five more knowledge resources beyond the script opposition. These additional knowledge resources are the logical mechanism (i.e., the resolution phase of the incongruity-resolution theories), the target (i.e., the person or group made fun of in the joke), the situation (i.e., the background assumed by the joke), the narrative strategy (the genre of the joke), and the language (i.e., the lexical, syntactic choices of the text) [11].

Since the focus of this paper is lexical semantics, we base our evaluation of WordNet solely on SSTH because the additional knowledge sources introduced by GTVH cannot be captured by the semantic relations between words. This limitation potentially reveals several distinct human abilities in understanding humor, as compared to lexical databases. We also want to note that although our input examples are constrained to puns (defined as "a type of wordplay in which similar senses or sounds of two words or phrases, or different senses of the same word, are deliberately confused"), we believe SSTH outlines the necessary and sufficient conditions for comprehending puns, as illustrated in the above example. Therefore, we do not modify SSTH in any way in our evaluation.

Ontological Semantics and ConceptNet

As described in the GTVH and detailed in Nirenburg and Raskin's book *Ontological Semantics*, Raskin and colleagues believe natural language processing tools must adopt the ontological semantic approach to be able to rigorously analyze texts such as jokes [12]. In such an approach, an ontology, or a constructed world model, is used to extract and represent meaning of natural language texts, reason about knowledge derived from texts, and generate texts based on representations of meaning.

This approach is lacking in WordNet, since it is primarily a lexical database instead of a lexical ontology. In contrast, ConceptNet is an ontological system for both lexical knowledge and common sense knowledge representation and processing. Consequently, ConceptNet contains many more kinds of relationships than WordNet, and is optimized for making practical context-based inferences over real-world texts [13].

We hypothesize that because ConceptNet is richer in its word-relation representation, it would perform better than WordNet in making connections among words/phrases and thus understanding humor. Benchmark results are presented in Section VI. Evaluation.

IV. Methodology and Implementation

We implemented two programs, one using WordNet and the other using ConceptNet, to analyze examples that conform to a particular format of jokes involving puns. We were interested in seeing whether semantic and ontological relations used in these two databases are sufficient for pun identification.

Input Examples

Jokes to be inputted into our program conform to the "Walk into a bar" format, which follow the following template:

```
A ___ walks into a bar.
The barman says, ``____''
```

Our program could be extended to handle other types of jokes, but we focused on one format for illustration. Examples of the "Walk into a bar" joke we are considering include:

- A tennis ball walks into a bar. The barman says, "Have you been served?"
- A horse walks into a bar. The barman says, "Why the long face?"
- A measle walks into a bar. The barman says, "Shots for everybody!"
- A beaver walks into a bar. The barman says, "Close the dam door!"
- A neutron walks into a bar. The barman says, "For you, no charge!"

The complete set of inputs that we used can be found in jokes.txt in the code folder.

Program using WordNet

Organization of words in WordNet is described in the previous section. Our program utilizes definition, examples, lemmas [1], and hypernyms for a word to construct its "association list."

The program identifies the pun based on matches between association lists of keywords and states its confidence level based on the strength of match.

Given the following input:

A goat walks into a bar. The barman says, "We don't serve any kids."

The program identifies the pun but is not entirely certain of it:

[1] A lemma is the grammatical/canonical form of a word that is used to represent the word in dictionaries (in this case, the "dictionary" is WordNet). A lemma specifies a single meaning for the word. The lemma set for a given word form is a set of lemmas providing all (word, meaning) mappings for it.

’kids’ is directly connected to the meaning of ‘goat’.
No Wordnet connection between ‘kids’ and ‘bar’ could be found.
The joke is probably funny because it makes a pun on the word ‘kids’. However, it’s uncertain.

Given another example input:
A tennis ball walks into a bar. The barman says, “Have you been served?”

The program identifies the pun and is certain of it:
‘served’ is connected to the meaning of ‘tennis_ball’ via ‘ball’ and ‘used’.
‘served’ is directly connected to the meaning of ‘bar’.
The joke is funny because it makes a pun on the word ‘served’.

The complete outputs for the WordNet analyzer (using jokes.txt as input) is in wordnet_main_outputs.txt.

Implementation

Our program reads a file of jokes into a list and iterates through it, processing every joke individually. For each joke, it determines a “customer” and a “statement”, i.e. what strings are filling in the blanks listed in the format above.

Currently there are a few customers that WordNet doesn’t recognize; for those we have hard-coded in a mapping for converting them to a WordNet-understandable form. For instance, we treat “measle” as “measles”. Additionally, we remove articles such as “the” and convert spaces into underscores so that WordNet will recognize the customer as a single word.

Then, we use WordNet to produce a list of words associated with the customer word. We do this by gathering all of the words in the definition, examples, lemmas, and immediate hypernym of the customer, removing any duplicates or stop words (semantically irrelevant words that we can ignore, such as “although”). We currently also have a hard-coded association list for the word “bar”.

Next, we return to the “statement” and try to determine a keyword which we suspect might be the pun. To do this, we go through all of the words in the statement, make an association list for that word, and then compare that word and its association list with the customer, bar, and their association lists. We then select the keyword that matches best. The strength of a match is determined by weighing the following matches differently: (1) a direct match with “bar,” (2) an indirect match with a word in the association list of “bar,” (3) a direct match with the customer, (4) an indirect match with a word in the association list of the customer. The direct matches contribute more to the program’s confidence level than indirect matches. Again, we remove any stop words.

Finally, we produce an output message saying what the pun was and how confident we are of the pun. We also state the type of connections in the pun and whether the connection was made through an intermediate word (i.e. there was a word shared in both association lists).

Program using ConceptNet

ConceptNet is a semantic network organized into a knowledge graph. In this network, words are represented as nodes, and the relationship between them is represented as edges. There are few different types of edges representing different types of relationships.

One of the key differences between ConceptNet and WordNet is there are no “senses” in the ConceptNet data. It does the bare minimum to distinguish different senses of a word—only using parts of speech tags. There is also no emphasis on synonymy as in WordNet—it is a relationship between words just like any other association. As a result, ConceptNet, unlike WordNet, cannot capture the nuance between different “senses” of a word.

Given this limitation, we aim to identify puns by looking for strong associations between keywords in the joke. The hypothesis we generated from ConceptNet are thus more of a likelihood analysis that tells us which parts of a sentence has a maximum likelihood of involving a pun. For instance, given the input:
A tennis ball walks into a bar. The barman says, “Have you been served?”

The program would output:
There is a pun involving:
served and drink
tennis ball and served

The complete outputs for the ConceptNet analyzer (using jokes.txt as input) is in conceptnet_main_outputs.txt.

Note that this program will not tell us exactly which senses of the word "served" are relevant to the pun. This is inherent in the ConceptNet system, and we cannot circumvent the issue without making significant inferences using some other database.

Finally, in contrast to the recursive lookups in WordNet, lookups in ConceptNet are direct because two concepts connected through a different concept have a direct connection between them as well. This property does not always hold in WordNet.

Implementation

Our program first identifies the keywords using the template. We used a regular expression matching to extract the unique information from each joke.

After that, we use the `nltk`'s associated stopword list to remove common words that are likely to have no bearing on the joke itself. Then, we tokenize the words by splitting them on spaces and punctuations. We noticed we could improve this process by properly dividing the sentence into phrases. Such an improvement is proposed as a next step.

Once this analysis is done, we get two sets, the subject set and the quote (barman's statement) set. In the example given above, the sets are {*"tennis", "ball"*}, and {*"served"*}, respectively. ConceptNet is capable of measuring the connection between these two sets. However, given the joke takes place in a specific context, we add another set of word, which is the "context set" for any given setting. For these bar jokes, our chosen "context set" was

{ *"bar", "drink", "alcohol", "bartender"* }

With these three sets, we analyzed the average weights of the relationships between words in different sets. We used the ConceptNet API to retrieve the information about the relationship between two words, specifically the weight of the relationship. In the example case, the average weight of the relationship between the subject set and the context set is -0.027, the average weight between the quote set and the context set is 0.09375, and the average weight between the quote set and the subject set is 0.083.

Using these average weights of relationships between different sets, we then determined which parts of the joke are involved in a pun. We did so by comparing the average relationship weight to a cutoff. In our program, the cutoff is set to 0.2 to exclude weak connections between words found in ConceptNet. Once we have dropped the inter-set relationship with the lowest weight, we can say with reasonable confidence that the wordplay involves the remaining sets of words. In our example, the hypothesis is there is a pun involving the context set and the quote, as well as the quote and the subject set.

```
There is a pun involving:
served and drink
tennis and served
```

This approach can be fine-tuned even more, but for the limited example input we have, it gives rather sensible hypotheses. The complete outputs for the ConceptNet analyzer (using jokes.txt as input) is in conceptnet_main_outputs.txt.

V. Testing

Following SSTH, for each joke we analyzed in WordNet or ConceptNet, we evaluated its understanding on the basis of the following questions:
(1) Is WordNet/ConceptNet getting multiple senses or meanings of the keyword?
(2) Is WordNet/ConceptNet detecting compatibility between each sense or meaning of the keyword with one script of the given text?

If the answers are positive for both questions, we categorize the result as "complete understanding." Otherwise, we consider WordNet or ConceptNet not capable of understanding the given jokes.

Ideally, we would also include a third criteria: Is WordNet/ConceptNet detecting oppositeness in the different scripts? However, this question is hard to answer for both databases as "oppositeness" does not have a clear definition; it is not as simple as antonymy. Therefore, we currently exclude this criteria in our testing analysis, though its addition is considered for future work.

WordNet

Testing of WordNet accuracy was done using the the above rubric. The output for the WordNet program is such that we can not only measure whether or not it correctly guessed what word the pun was on, but also whether or not it "understood" the dual connections in the pun. As shown in Results, the output allows us to see whether multiple senses are found for the keywords and whether each identified sense is compatible with a script.

Out of the 15 inputs, the WordNet program correctly guessed the keyword from the bartender's statement 8 times. Additionally, both times when there were two keywords (i.e. "long face" or "serve food"), it correctly guessed one of those words. However, it only understood the correct reasoning for a connection 8 times out of a potential 29 (one joke was not a pun, and thus had only one connection instead of two). It also only completely understands the joke 1 out of 15 times (it is successful for the tennis ball joke).

This number could potentially be larger if not for some instances of false positive connections crowding out the true keyword in the statement (e.g. In "A tree walks into a bar. The barman says, 'I think you would better leave.'", the program could identify a connection between "bar" and "leave", but this is ignored in favor of false positives on "better"). Nevertheless, the relatively low score shows that sometimes the program reaches the correct result by chance. Rather than using real understanding via WordNet, it may be correct via a "right for the wrong reasons" false positive or via the joke having only one word in the barman's statement that is not a stop word.

The following are some notable issues we discovered from testing the WordNet program:

- WordNet does not have any section for the pronunciation of words, and thus, it cannot be used to analyze homophone-based puns, (e.g. "leave" and "leaf", "dam" and "damn").
- WordNet does not understand most idioms, or almost any combination of multiple words. "Long face", and "off the wagon" are meaningless to it.
- WordNet lacks entries for some relatively common words. "Measle", "E-flat" and "Comic Sans" do not have respective entries. Sometimes we can manually find a similar word, but in the case of the joke, "E-flat walks into a bar. The bartender says, 'Sorry, we don't serve any minors,'" "E-flat", switching to "flat" produces a lot of extra word senses that are irrelevant.
- WordNet is ultimately not made for determining associations between words, and it will often miss connections that are not extremely close. It misses the connection between words such as "bar" and "minors", "amoeba" and "split", as well as "bar" and "charge". Surprisingly, it even missed the connection between "sandwich" and "food", due to a quirk where the hypernym hierarchy of "sandwich" goes from "snack_food" to "dish" to "nutriment" to "food". Notably, it would be infeasible to include 4th-level hypernyms within a word's association list; consider the fact that the 4th-level hypernym of "food" is the rather unrevealing "entity."

ConceptNet

Performance of WordNet is benchmarked against that of ConceptNet, using the same inputs and criteria. Once the ConceptNet program generated hypotheses for all inputs, we manually went through the results to see which associations made by ConceptNet capture the pun in and which associations are not as funny or central in understanding the joke.

Out of the 15 inputs, it correctly identifies 13 out of the 29 connections, and completely understands 3 out of the 15 jokes (it understands the tennis ball, measle, and neutron jokes). However, it notably fails to parse the regex in 1 case, which is due to the structure of the analyzer rather than due to the capability of ConceptNet, which could likely identify 1-2 additional connections, possibly identifying the entire joke, given a correct parse. It also generates 10 false connections; though, note that unlike in the WordNet analyzer, false connections aren't inversely proportional to real connections.

Since associations in ConceptNet lack hierarchical structure, it is impossible to determine the exact relations between the keywords that are used in the pun (e.g. while WordNet can find that "serve" has a meaning of "put a ball into play" and that "ball" is a hypernym of "tennis ball," ConceptNet can only tell us "serve" and "tennis ball" are related concepts). Therefore, the outputs of the program are not as telling as those from the WordNet program.

By testing the ConceptNet program, we note the following issues:

- The phrase division was not optimal. For example, ConceptNet finds a connection between "horse" and "long face", bot no connection between "horse" and the set {"long", "face"}. As a result, it failed to capture the joke in *A*

horse walks into a bar. The barman says, "Why the long face?", but could have captured it if we manually made the quote set {"long face"}.
- The cutoff to exclude weak word connections is important, but needs further tuning. We know that there should be a cutoff so that we do not return false positives, but the present cutoff leads to some false negatives. For example, in the example of the neutron joke (*A neutron walks into a bar. The barman says, "For you, no charge!"*), the code finds a pun between

  `neutron and bar`

  which may be too far fetched. Perhaps an adaptive cutoff system would be better—a system that bases the cutoff on how relevant the keywords are, and/or how strong the baseline connections are among all words in the given sentence.
- a system that leaves out more irrelevant common words (a better implementation of the "stop words") will improve pun recognition. Doing so removes noise as ConceptNet searches for associations, leading to more accurate output. The same point applies to WordNet.
- The ConceptNet analyzer fails to understand any jokes that do not fit into the regular expression, for example, *A duck walks into a bar with a bunch of friends, but all his friends ditch him. The barman says, "I guess the bill's on you."*. A generalized semantic meaning analyzer would let us extract better features out of this jokes, so we could actually understand what truly constitutes a keyword and what does not.
- Average weights of relationships between the context set, the quote set, and the subject set seems a useful metric, but the average sometimes misses important cases, for example, when the pun is centered on a single word instead of many. There is a trade-off because if we take the maximum of the weights, then a coincidental noise might dominate the pun recognition and thus lead to wrong results.

VI. Evaluation

Using the criteria listed in Testing, both the WordNet and the ConceptNet analyzers experienced some success at analyzing jokes. The WordNet analyzer fully understood 1 joke and correctly found 8 connections, whereas the ConceptNet analyzer fully understood 3 jokes and correctly understood 13 connections.

Additionally, one of the crucial tasks in the WordNet analyzer was to create association lists for a particular word, which is basically what ConceptNet is already designed to do. Thus, ConceptNet is definitely better suited for the task of recognizing puns, as using it in a program does not require the same pre-analysis work. These findings confirms our hypothesis.

ConceptNet is also more reliable in that it is unlikely to miss basic connections (e.g. connections between "sandwich" and "food") as it associates concepts directly instead of via hierarchical organization. However, ConceptNet still sometimes missed associations where WordNet succeeds (e.g. the WordNet analyzer notices a connection between "horse" and "face" while the ConceptNet analyzer did not).

Our analysis reveals lexical semantics in WordNet is indeed representative of human lexical memory in some instances, but it is far from a complete representation. In line with previous work on ontological semantics, rich ontology in ConceptNet demonstrated more success in analyzing jokes. Consequently, we believe should WordNet contain a ontological system as an extension (e.g. with reference to ConceptNet) and its corresponding analyzer fight off false positives more aggressively, it would be more successful in humor recognition. Such combined approach would be better than using ConceptNet alone, because the organization of words in WordNet provides more insights into how two keywords are related (as we can trace the hypernymy hierarchy, for example). This feature naturally translates into a sensible reasoning for why a given joke or pun may be funny, which better approximates what humans can do.

One of the most surprising facts about this project was how easy it was to guess puns just by process of elimination, i.e. via removing "stop words." For example, the WordNet analyzer correctly evaluated the non-pun joke ("A rabbi, a lutheran minister, and a priest walk into a bar. The barman says, 'What is this? A joke?'"), identifying "joke" as the keyword, just via the fact that everything else in the barman's statement was semantically irrelevant.

Despite the success of our relatively simple programs in humor recognition, semantics in lexical databases such as WordNet (and ConceptNet) are still fundamentally lacking compared to humans. In particular, both SSTH and GTVH call for a complete understanding of the script, or the situation, which is almost impossible to construct solely based on semantic relations in lexical databases. Moreover, GTVH require extra knowledge sources such as the target, the narrative strategy, the language etc., which are intuitive to humans but not at all to programs relying on these databases. This is an area for improvement if such lexical databases aim to better simulate the psycholinguistics of humans; namely, determining the situation and possibly additional implied features of the given text based on the lexical forms of individual words.

VII. Conclusion

Program Improvements

Our implementation of the WordNet and ConceptNet analyzers are relatively simple and can benefit from the following three improvements.

First, both programs would reduce false positive rates if we eliminate irrelevant words more aggressively. This may be done through neural networks that train the model to sort words in inputs from most to least relevant in the context of jokes with a bar setting.

Second, SSTH requires an additional criteria that is currently not tested in our programs: Is WordNet/ConceptNet detecting oppositeness in the different scripts? Incorporating this criteria would also help us decrease the amount of false positives, as the keywords are only essential when this sense of "oppositeness" is present. To do so, we first need to clearly define what "oppositeness" entails and then utilize the existing word relations in the two databses to best capture it.

Finally, as mentioned in a notable issue we observed when testing the ConceptNet analyzer, an adaptable cutoff to exclude weak word connections is preferred over a hard-coded cutoff.

WordNet Extensions

Compared to ConceptNet, WordNet is quite limited in its ontology. Word realtions are too sparse in WordNet for the application of humor recognition.

To make WordNet better suited for processing practical and context-oriented text, it is desirable to extend it with a database such as ConceptNet, so that it can take advantage of informal, commonsense knowledge that are not encoded in its own database.

Project Extensions

There are several ways we would seek to extend our project if we had significantly more time or a significantly larger team.

One of the first things we could do is to collect more jokes, and expand our programs to handle more variations within a joke. Slight alterations to the joke template each require extra code. For example, when the barman "shouts" instead of "says", that required a special modification to the WordNet analyzer. Being able to handle more of these special cases would allow us to increase the number of inputs we have and would let us handle those inputs better.

An even more significant expansion would be to handle more types of joke templates. For example, during our preliminary analysis, we looked at other structured jokes, such as Yo Momma jokes. An example would be, "Yo momma's so fat, she uses the carpet as a blanket!" These used the format:

Yo momma is so fat, she [verb phrase].

Additionally, we looked at some image memes, such as "Bad Luck Brian" in which a good or neutral incident is twisted to have the worst possible interpretation. An example would be, "Invited to play a game – Russian Roulette" These used the format:

[something happens] – [negative instance].

With more joke examples and formats, we can identify more semantic relations that WordNet does not capture adequately in its current design, postulate why they are not captured, and make suggestions for how to incorporate them into WordNet.

For example, our preliminary analysis of Yo Momma jokes suggested that it would be useful if WordNet captured words' properties, such as size, texture, color, etc. In most Yo Momma jokes, there are two nouns with similar properties, but where one noun is significantly bigger than the other (e.g. "carpet" and "blanket" are both soft, flat coverings, but a carpet is much larger than a blanket.) Having a "properties" section would thus allow WordNet to handle more instances of humor. Currently ConceptNet actually has a "properties" section, but it seems much smaller than it should be. For example, the only property of "blanket" that ConceptNet lists is "dark". Thus, it seems like there is plenty of room for improvement in both WordNet and ConceptNet for semantic understanding, which could be usefully discovered via the analysis of more joke formats.

Another route to improve our program would be to generate example jokes to confirm our analysis. As one might imagine, if the success/failure patterns recognized by our program are highly descriptive of the jokes, then we can use semantic relations that WordNet is good at establishing to generate "jokes" that are humorous most of the time. By looking at the generated jokes, we can get a sense of how accurately we have analyzed and assessed lexical semantics in WordNet.

However, it is unlikely that we would be able to easily programatically generate puns such as those in the "walk into a bar" format. Rather, we had determined in our earlier analysis that WordNet would be fairly good at generating anti-jokes of the form:

I'm not just [negative adjective or noun phrase]. I'm also [other negative adjective or noun phrase].

There are also variants on this structure, such as:

I'm not just a [negative noun phrase]. I'm an [adjective] [same negative noun phrase].

I'm don't just [negative verb phrase]. I also [other negative verb phrase].

Some examples are:

- I'm not just lazy. I'm also crazy!
- I'm not just a spoiled princess. I'm a rich spoiled princess.
- I'm not just an idiot. I'm an American idiot.
- I don't just get C's. I also get F's.
- I'm not just single. I also have no friends.

This structure is humorous because it subverts the listeners expectations. Rather than disagreeing with the insult and protest that they have some valuable attribute, the joker instead self-deprecates themself further. Or in some cases, the joker attempts to argue that they have a valuable attribute, but obliviously makes themself seem even worse to the listener.

Essentially, we hypothesized that we could make use of WordNet's ability to easily find synonyms or hypernyms of words to make a second, similar sentence, given the structure of the first one.

Final Thoughts

Inspired by the theories of psycholinguistics, WordNet has been improved over the years to better simulate how humans approach lexical semantics. However, the current design of the system is still short of a complete representation of human lexical memory. In our project, we explored the strengths and shortcomings of WordNet in its lexical semantic analysis via the lens of humor understanding. We found although the structured organization of words in WordNet is advantageous in explaining the exact relationships between two words, it limits the relational ontology and thus renders WordNet less competent in processing practical, context-oriented texts when compared to ConceptNet.

To improve its ability in such tasks, we can extend WordNet with informal, commonsense knowledge as provided in ConceptNet. Nonetheless, it is likely that an optimized program using both databases would still miss jokes that appear obvious to us. Specifically, it seems a purely human ability as of now to quickly and intuitively identify knowledge sources such as script opposition, logical mechanism, the target, the situation, the narrative strategy, and the language. We believe these abilities are precisely the unique aspects of understanding humorous discourse that remain a major challenge in the realm of computational humor.

References

[1] G. Miller, R. Beckwith, C. Fellbaum, Derek Gross, and Katherine Miller. Introduction to WordNet: An On-line Lexical Database, *International Journal of Lexicography*, Volume 3, Issue 4, 1 December 1990, Pages 235244, https://doi.org/10.1093/ijl/3.4.235

[2] Ferreira, B. (2016). Joke-Telling Robots Are the Final Frontier of Artificial Intelligence. Motherboard, Vice. motherboard.vice.com/en_us/article/z43nke/joke- telling-robots-are-the-final-frontier-of-artificial-intelligence

[3] Taylor, Julia M. "Computational detection of humor: A dream or a nightmare? the ontological semantics approach." *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03.* IEEE Computer Society, 2009.

[4] G. Miller. Nouns in WordNet: A Lexical Inheritance System, *International Journal of Lexicography*, Volume 3, Issue 4, 1 December 1990, Pages 245264, https://doi.org/10.1093/ijl/3.4.245

[5] C. Fellbaum, D. Gross, and K. Miller. Adjectives in WordNet, *International Journal of Lexicography*, Volume 3, Issue 4, 1 December 1990, Pages 265277, https://doi.org/10.1093/ijl/3.4.265

[6] C. Fellbaum. English Verbs as a Semantic Net, *International Journal of Lexicography*, Volume 3, Issue 4, 1 December 1990, Pages 278301, https://doi.org/10.1093/ijl/3.4.278

[7] R. Beckwith., G. Miller, and R. Tengi. Design and Implementation of the WordNet Lexical Database and Searching Software. In WordNet. An Electronic Lexical Database (1998), pp. 105-127

[8] S. Castro , M. Cubero, D. Garat, and G. Moncecchi. (2016) Is This a Joke? Detecting Humor in Spanish Tweets. In: Montes y Gmez M., Escalante H., Segura A., Murillo J. (eds) Advances in Artificial Intelligence - IBERAMIA 2016. IBERAMIA 2016. Lecture Notes in Computer Science, vol 10022. Springer, Cham

[9] R. Mihalcea and C. Strapparava. Making Computers Laugh: Investigations in Automatic Humor Recognition, University of North Texas Libraries, digital.library.unt.edu/ark:/67531/metadc30968/

[10] Hempelmann, Christian, Victor Raskin, and Katrina E. Triezenberg. “Computer, Tell Me a Joke... but Please Make it Funny: Computational Humor with Ontological Semantics.” *FLAIRS Conference. Vol. 13.* 2006.

[11] Attardo, Salvatore. “Semantics and pragmatics of humor.” *Language and Linguistics Compass 2.6 (2008)*: 1203-1215.

[12] Nirenburg, Sergei, and Victor Raskin. *Ontological semantics.* Mit Press, 2004.

[13] Liu, Hugo, and Push Singh. “ConceptNeta practical commonsense reasoning tool-kit.” *BT technology journal 22.4* (2004): 211-226.