Toward Computational Recognition of Humorous Intent

Julia M. Taylor (tayloj8@email.uc.edu)
Applied Artificial Intelligence Laboratory, 811C Rhodes Hall
Cincinnati, Ohio 45221-0030

Lawrence J. Mazlack (mazlack@uc.edu)
Applied Artificial Intelligence Laboratory, 811C Rhodes Hall
Cincinnati, Ohio 45221-0030

Abstract

Recognizing the intent of a written utterance is a difficult task. To understand the intended meanings, it may be necessary to recognize all possible meanings of the utterance; and, then choose the most appropriate one for the situation. The choice may be made using heuristics. A method for recognition of a humorous intent in a text is proposed. Computational recognition of a humorous intent can be broken down into two parts: recognition of a humorous text, and recognition of the intent to be humorous. To narrow the focus, we propose to recognize the humorous intent of short dialogs. A dialog will be considered humorous if the first part of the text can have two meanings; and, one of the meanings conflicts with the meaning of the punchline. The intent of the text is considered humorous if the schema related to the non-conflicting meaning of the setup has not been activated in the preceding text.

Keywords: computational humor; intent recognition; jokes; wordplay.

Introduction

Natural language processing is a difficult and increasingly important topic to study. With computers becoming more and more "knowledgeable" and assisting in some everyday tasks, the need for an accurate natural language processing system becomes more apparent and desirable.

Recognizing the intent of a written utterance is a difficult task. To understand the intended meaning without additional hints, it may be necessary to recognize many meanings of the utterance and then choose the most appropriate one for the situation. The choice may be made using heuristics.

It is not always an easy task for a human, let along the computer, to recognize the meaning of an utterance. One difficulty is because the utterance can have both a literal and a non-literal meaning. For example, the literal meaning of $Text_1$ is "Are you capable of closing the door?" while the indirect meaning is "Please close it." Most people, through their experience, know that the intended meaning of $Text_1$ is indirect.

$Text_1$: Can you close the door?

However, if one says: "It is cold here," the intended meaning may be "Please close the window." Understanding this intended meaning is not trivial.

Disambiguation is not only a matter of choosing between literal and non-literal meanings; there also problems with words having alternate meanings and alternate parsing caused by either competing sentence structure (see Figure 1) or words with alternative meanings, for example, "fly' being a noun and a verb.

John likes (ham and eggs).
John likes (ham) and (eggs).

Figure 1: Competing sentence structure

There are different theories of indirect or figurative language comprehension. "A figurative language is language that means one thing literally, but is taken to mean something else" (Carroll, 2004). One of the figurative language theories is a pragmatic theory. The pragmatic theory (Searle, 1975; Carroll, 2004) states that figurative language is being comprehended by first considering the literal meaning of the utterance and then rejecting it if the listener decides that the literal meaning was not intended.

This paper will concentrate on the task of recognizing if a text is intended to be humorous. In other words, can a text be a joke? If it can, was it intended to be a joke?

Many texts that appear on the Internet are semantically tagged. The tags are invisible to a human, but readable for a computer. Some tags provide information about the intentional topic of the document. For example, if a title of a web-page tagged with "jokes", it is likely that this web-page contains humorous texts. Similarly, if the title of the web-page is tagged with "world news", it is less likely that most of the text is humorous.

For the purposes of humorous intent recognition, the documents (or texts) could be divided into three categories:

(1) Documents containing a semantic tag indicating that the text is intended to be humorous, regardless of the actual funniness level of the text.
(2) Documents containing a semantic tag indicating that the text is intended to be non-humorous.
(3) Documents that do not contain any tag that would provide clues to the intended humorous level of the text.

This paper will concentrate on recognizing the intent of sentences in categories (2) and (3). The intent of texts in the category (1) can be recognized by reading the tag. The intent of the text in category (2) could also be recognized by reading the tag. But, this category is still of interest as the texts may contain humorous sentences.

A method similar to pragmatic theory will be used to recognize humorous intent of an utterance or a text. Parts of the candidate text will be examined to find their most frequently intended meaning (the literal meaning in pragmatic theory). If this meaning does not agree with the rest of the candidate text, a different meaning that agrees with the rest of the text will be explored. The candidate text will be considered humorous if both of the following conditions hold:

- The first, most common, meaning of the first part of the text does not agree with the rest of the text.
- There is another meaning of the first part of the text that does agree with entire text.

To computationally recognize a humorous text, several things must be present. First, a computer should at least partially understand natural language, and activate alternative possible meanings of a grammatically structured text. Secondly, a system should have access a database of world knowledge to determine if the text is meaningful. To determine if a humorous text was intended to be humorous, previous context has to be considered.

Not all humorous sentences are intended to be humorous. Consider $Text_2$ as example:

> $Text_2$: "It was mentioned on CNN that the new prime number discovered recently is four times bigger than the previous record."
>
> John Blasic

$Text_2$ is funny if one knows that a prime number cannot be divisible by anything but one and itself. Therefore, a number that is four times larger than a different number, cannot be prime. However, it is unlikely that the CNN reporter intended it to be funny.

Jokes

Humor is a part of natural language. Verbal humor is a subset of all types of humor. It results from incongruous texts or dialogs, where the resolution to incongruity is provided in the final sentence. Sometimes incongruity is caused by ambiguity in the text. To understand the ambiguity, and disambiguate the text, both natural language understanding and world knowledge is required.

A subset of verbal humor is the joke. A joke is a short humorous narrative where the funniness culminates in the final section (Hetzron, 1991). Many humor theorists agree that a joke typically consists of a setup and a punchline. The setup is the part of the joke that prepares a reader's expectations.

The punchline breaks the expectation created by the setup (explicitly or implicitly stated). When the setup of a joke is read, the reader comes up with an interpretation of the setup ($Interpretation_1$). When the punchline is read, the meaning of it conflicts with $Interpretation_1$ of the setup, causing the reader to come up with a different interpretation ($Interpretation_2$), which does not conflict with the setup. (see Figure 2)

Consider $Joke_1$ as an example.

> $Joke_1$: Customer: Do you mind if I try on that dress in the window?
> Sales Assistant: Wouldn't it be better to use the fitting room?

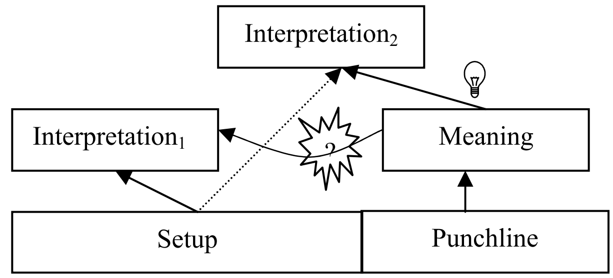

Figure 2: Joke description (Ritchie, 2004)

In $Joke_1$, the setup is: "Customer: Do you mind if I try on that dress in the window?" The first, more obvious interpretation of the setup is that the customer wants to try on a dress that is located in the window. The punchline, "Wouldn't it be better to use the fitting room," reveals the second, conflicting, interpretation: customer wanting to change their clothes in the window. The first interpretation conflicts with the punchline, while the second one makes sense, even though it is an odd thing to do. "The comic effect arises when an alternative, non-favored and therefore non-expected interpretation is revealed, at the punchline, as the correct one" (Dascal, 1985; Ritchie, 2004). Theories that favor incongruity and resolution "explanation" of jokes are called Incongruity-Resolution Theories.

Raskin's (1985) Semantic-Script Theory of Verbal Humor (SSTH) can be considered an Incongruity-Resolution theory. Raskin argues that there are two necessary and sufficient conditions for a text to be humorous:

- A text has to be compatible, fully or in part, with two different scripts.
- The two scripts with which the text is compatible are opposite, and must overlap fully or partially.

Raskin defines a script as "a large chunk of semantic information surrounding the word or evoked by it. The script is a cognitive structure internalized by the native speaker and it represents the native speaker's knowledge of a small part of the world."

We can say that a text *can* have a humorous intent if the text compatible, fully or in part, with two scripts that overlap and oppose.

To find all the possible scripts used in different texts, a complete natural language understanding is required. Complete Natural Language Processing (NLP) is far from being achieved.

Wordplay Jokes

Recognition of all verbal humor is an overly broad topic. To narrow down the focus, a subclass of verbal humor is considered. Wordplay jokes are jokes that depend on words that are similar (or the same) in sound, but have two different meanings. The setup promotes one meaning. The punchline unveils another meaning of a word in the setup, or a meaning of a word that sounds similar to a word in the setup. The difference between the two meanings creates a conflict by breaking expectations. And, by the previous definition, is humorous.

Wordplay can be created between two words with the same pronunciation and spelling, with two words with different spelling but the same pronunciation, and with two words with different spelling and similar pronunciation. For example,

> $Joke_2$: Nurse: I need to get your weight today.
> Impatient patient: Three hours and twelve minutes.

The text is a joke due to the "confusion" between *wait* and *weight*. From the SSTH point, the joke has two scripts, "waiting time" and "person's weight," that overlap in pronunciation of "*wait/weight*" and differ in meaning. From the intent recognition method, the most common meaning for the first sentence is "I need to get your *weight* today" as it is common for a nurse in the doctor's office to measure a patients' weigh. The most common meaning of the first sentence does not agree with the second sentence. Another meaning of the first sentence is "I need to get your *wait* today". The second meaning is unlikely, but, perhaps, the office is trying to improve patients' waiting time. The second meaning of the first sentence agrees with the second sentence. Therefore, the text is a joke; and, could be intended to be humorous.

Communication and Discourse

Any computational intent recognition is, at the very least, as difficult as intent recognition in conversations between two people. As stated in the introduction, it is not easy to recognize the intent of all utterances in conversation between two people. What makes the intent recognition difficult is our inability to "read" somebody else's mind. To put it in other words, we do not always understand the intended meaning of a written or spoken text.

This paper focuses on recognizing the humorous intent of dialogs. Dialogs can be in written form as well as oral. For example, online chat sessions such as AOL's instant messenger can be looked at as real-time dialogs. For this dialog to be successful, people have to understand the intended meaning of the phrase as opposed to its literal meaning. Most of us have read a sentence in an email, or heard something in a conversation, and wondered what was actually meant by it.

Discourse comprehension is more difficult than sentence comprehension. Discourse comprehension "depends less on the meaning of the individual sentences than on their arrangement. Indeed, it is entirely possible for a group of meaningful sentences to be thrown together in a way that makes no sense at all." (Carroll, 2004)

For a person to understand a joke, the joke has to be coherent and relevant to their world knowledge. Coherence is defined as "the range of possibilities that exist for linking with what has gone before" (Hasan & Halliday, 1976). In the context of this paper, world knowledge is a mental model. In other words, it is "a cognitive structure that represents some aspect of our environment" (Carroll, 2004). Therefore, we can say that jokes are funny if we find them coherent and incongruous according to our mental model. The resolution of the incongruity has to correspond to our mental model as well.

For a text to be coherent, all information does not have to be present in the text itself. Some can come from a mental model. For example (Carroll, 2004):

> $Text_3$: John bought a cake at the bake shop. The birthday card was signed by all of the employees. The party went on until after midnight.

The sentences are not syntactically connected, yet we seem to make sense out of them and make them into one story.

Sometimes it is difficult to understand the text without an appropriate activation of a schema of a mental model. "A schema is a structure in semantic memory that specifies the general or expected arrangement of a body of information" (Carroll, 2004).

Schema activation can play an important role in jokes. Suppose we have a conversation about athletic organizations for kids. It is likely that the schema for *athletic clubs* has been activated. When we hear the question "Do you believe in clubs for young people?" we think about clubs as organizations since this is the schema that is active. Suppose the conversation was not about athletic organizations, but about child abuse. Upon hearing the same sentence, it is likely that a different schema will be activated, if it hasn't been already. And, it would have something to do with hitting children with some objects. Now, consider $Joke_3$:

> $Joke_3$: --Do you believe in clubs for young people?
> --Only when kindness fails.

It could be argued that if the conversation was about child abuse, there is a chance that the "*Athletic organization*" schema was not activated. If the schema of hitting children is activated first, the first meaning of the first sentence will agree with the second sentence. Then, there would be no need to look for the second meaning of the first sentence. Would $Joke_3$ be considered a joke?

Syntactic Ambiguity in Jokes

There are jokes that depend on syntactic ambiguity. (Attardo et al., 1994) These jokes are based on the way we "group" words in a sentence. As an example, consider $Joke_4$:

> $Joke_4$: One morning I shot an elephant in my pajamas. How he got into my pajamas I don't know.[1]

The joke works because "I shot an elephant in my pajamas" can be interpreted as:

- I shot the elephant that was wearing my pajamas.
- I shot the elephant and I was wearing my pajamas.

"Parsing is the process of assigning elements of surface structure to linguistic categories" (Carroll, 2004). There are different strategies to parsing. Depending on what type of strategy is used, different trees will be constructed.

---

[1] Groucho Marx, *Animal Crackers*, 1930

Different strategies use different computational algorithms to choose grammar rules to build a tree. A sentence can be considered syntactically unambiguous if all parsing algorithms produce the same tree. A sentence can be considered grammatically correct if at least one tree could be built.

A syntactic ambiguity of one part of a text does not imply that the text is a joke. The candidate text with a *syntactic* ambiguity in the setup will only be considered a joke if *syntactic* ambiguity leads to *semantic* ambiguity, and the *semantic* ambiguity is resolved in the second part of the text, the punchline. For example, $Text_5$ (Carroll, 2004) is structurally ambiguous, but is not a joke:

$Text_5$: The boy hit the girl with the boomerang.

Computational Understanding of Natural Language

Natural language understanding is a complicated process. To fully understand a speaker, the listener has to comprehend the intended meaning of what the speaker has said. The earlier section on *Communication and Discourse* discussed how the context of the previous conversation or text effects schemata activation for the text that is being processed by humans. Can computers do it? Can computers fully process natural language?

The following quotes are taken from Government Computer News of June 23, 2004:

- Amtrak has installed speech recognition software to replace the button-pressing menus that drive many people mad. Now you can talk to a virtual salesperson named Julie to get train schedules, make reservations, pay for tickets, and discuss problems. Customers are happier, and Amtrak is saving money.
- IBM has a Super Human Speech Recognition Program to greatly improve accuracy, and in the next decade Microsoft's program is expected to reduce the error rate of speech recognition, matching human capabilities.
- MIT is planning to demonstrate their Project Oxygen, which features a voice-machine interface. Project director Rodney Brooks says, "I wanted to bring the machine into our world, a machine that will [...] let you ask questions in casual English, and answer them the same way."
- General Motors OnStar driver assistance system relies primarily on voice commands, with live staff for backup; the number of subscribers has grown from 200,000 to 2 million and is expected to increase by 1 million per year.
- The Lexus DVD Navigation System responds to over 100 commands and guides the driver with voice and visual directions.
- Reliable speech recognition should be common by 2010.

It seems like the interest in speech (and, therefore, natural language) processing is increasing. If reliable speech recognition can be achieved by 2010, recognizing humorous statements would be desirable.

It is not clear how to achieve complete natural language understanding. One approach could be to use ontologies.

"An ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information (a domain is just a specific subject area or area of knowledge, like medicine, tool manufacturing, real estate, automobile repair, financial management, etc.). Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them (note that here and throughout this document, definition is not used in the technical sense understood by logicians). They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable.... Ontologies are usually expressed in a logic-based language, so that detailed, accurate, consistent, sound, and meaningful distinctions can be made among the classes, properties, and relations." (W3C Recommendation [10 February 2004]).

It is speculated that ontologies can aid natural language processing. There are no ontological structures yet that can fully handle natural language. However, there are some ontologies that may be close to it.

There are a large number of existing ontologies. One of the largest, and probably most complete, is Cyc (Lenat, 1995). Cyc was intended to capture and represent knowledge in a context addressable form.

> "The Cyc knowledge base (KB) is a formalized representation of a vast quantity of fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. The medium of representation is the formal language CycL. The KB consists of terms — which constitute the vocabulary of CycL — and assertions which relate those terms. These assertions include both simple ground assertions and rules."[2]

Cyc can be used as a natural language processing system. The background knowledge captured in its knowledge base can be used to come up with the exact meaning or word in a sentence, even if the word can otherwise have more than one meaning otherwise.

Computational Recognition of Humor and Humorous Intent

Computational recognition of a humorous intent can be broken down into two sub-problems:

- Can the candidate text be humorous?
- Is the humorous text intended to be humorous?

To summarize previous sections, a candidate text is a joke if it is compatible with two different scripts that oppose. In other words, if a part of the text can have two interpretations, but only one interpretation works with the rest of the text.

[2] http://www.cyc.com/cyc/technology/whatiscyc_dir/whatsincyc

Intent Recognition

Once it is determined that the candidate text is a joke, it has to be decided if the text was intended to be humorous. According to Ritchie's (2004) description of a joke, a joke has an obvious and a hidden meaning. The obvious meaning conflicts with the punchline, and the hidden meaning does not. If the text was intended to be a joke, the first, most probable, meaning of the setup will conflict with the punchline, while the less probable meaning will not. The key is to find which meaning is most probable.

For humans, the choice of interpretations depends on the schemata activated by previous context. For a computer to simulate human behavior in humorous intent recognition, previous context must be considered in recognizing intended meaning of a candidate text as well.

As NLP is far from recognizing most common meanings, or overall meanings of text, this paper will assume that all texts of interest are semantically tagged. With the continuing development of Semantic Web, this assumption may not be far from reality. For the purposes of this paper, it will be assumed that each paragraph has a field that contains the key concepts that the paragraph talks about. These concepts will activate schemata. It should be noted that Semantic Web does not automatically assign key concepts to paragraphs.

It will also be assumed that words in the sentences may be semantically tagged. This means that a computer could potentially recognize the intended meaning of the sentence. However, since the tags can be seen by a computer, but not by a human that is reading the texts, these sentences would still be checked for other possible meanings. In this case, if a joke was found in an intentionally non-humorous text (according the meaning of the sentence received from the semantic tags), the sentence could be "flagged" to the author of the text to avoid potential misunderstanding and embarrassment. This case will be ignored for the rest of the discussion.

If the text can be a joke, and the previous context did not activate a schema relevant to the setup, it will be assumed that the speaker intended to tell a joke.

If the text can be a joke, but the last schema activated by the previous context is needed for the hidden interpretation of the joke, it will be assumed that the text was not intended to be humorous. A better version of this approach is to use conditional probability to calculate which interpretation is activated first, given the schemata of the previous context. This means that $Joke_3$ would not have a humorous intent if the last topic was about hitting children and its schema was activated before "*athletic organizations*."

This also means that if a company finds a potential joke in an email, it will not block the email if the hidden interpretation of the joke closely relates to the previous discussion; and, the previous discussion activates hidden interpretation faster than the obvious interpretation.

Joke Format

The domain of wordplay jokes is large. To restrict the problem, only two lines dialogs containing wordplay could be considered. Each line contains only one sentence. The first line of the dialog is a question, and the second line of the dialog contains an answer to the question. The question line is the setup of the joke; the answer line is the punchline. Two meanings of the setup should be based on wordplay. The punchline should be based on one meaning of the setup. It will be assumed that the punchline is based on the *hidden* meaning of the setup.

Using these rules, $Joke_3$ will be considered as a candidate text for a joke; but, $Joke_2$ will not be, as the first sentence is not in the form of a question. $Joke_2$ could be rewritten as $Joke_{2.a}$.

$Joke_{2.a}$: Nurse: Can I get your weight today?
Impatient patient: Three hours and twelve minutes.

$Joke_{2.a}$ will be considered as a joke candidate because the first line is a question. The first line contains wordplay. Both meanings of the first line are based on the wordplay. The second line is the answer to the question asked in the first line. The second line is based on one of the meanings of the first line.

It is possible to restrict the problem even further. Only questions relating to certain subjects could be used. For example, only math questions can be used in the first line. If that is the case, $Joke_{2.a}$ will not be considered.

A restriction to a particular domain (in this case mathematics) narrows down the number of concepts and concept relationship that the computer must understand in order to recognize jokes.

Wordplay

It is easy for a human to see that the first line of $Joke_{2.a}$ contains wordplay: *wait*/*weight*. Computational recognition of wordplay starts with comparing orthographic similarity of tags in the setup and punchline. The two tags that are orthographically most similar will become wordplay candidates. Then, repeating substitution of letters, guided by a heuristic approach, will be used to transform one word into another. If such transformation can be made, the word in the setup will be substituted with its wordplay. If the setup containing wordplay is syntactically and semantically correct, the text will be considered a joke. Syntactic correctness can be verified by a parser, semantic correctness can be verified by using the ontology and CycNL.

Consider $Joke_{2.a}$ as an example. Each verb, noun and adjective is semantically tagged. A noun phrase, verb phrase or a prepositional phrase can be tagged as well. In $Joke_{2.a}$ "*three hours and twelve minutes*" is tagged as "*wait*".

The orthographic similarity between concepts can be calculated using the LCSR coefficient. LCSR is calculated by dividing the length of the longest common subsequence by the length of the longest string. The similarity of "*wait*" and "*weight*" equals the length of "*wit*" divided by the length of "*weight*," which is 0.5. This similarity is the highest among all possible setup/punchline pairs. Letters in "*weight*" will be substituted with similar-sounding letters, until the word "*wait*" is produced; or, all possible substitutions are made and but we do not get "*wait*".

If "*wait*" is found, "I need to get you weight today" and "I need to get your wait today" will be syntactically and semantically verified. If both are successful, the dialog is a joke.

If "*wait*" is not found, a setup/punchline concept pair with the second highest similarity will be considered.

Parsing

A syntactic structure of each sentence will be validated through a parser. If a candidate text contains a sentence that does not follow the grammatical rules, the text will not be considered humorous. A sentence does not follow the rules of grammar if the parser cannot build at least one parse tree for the sentence. It is assumed that the parser uses algorithms based on several strategies, and builds more than one parse tree for ambiguous sentences.

If the setup sentence is syntactically ambiguous, each parse tree will be used to find semantic interpretations of the sentence. A sentence can be syntactically ambiguous and contain a wordplay if the wordplay is based on words with the same spelling. The text will be considered a joke if the punchline does not conflict with the meaning of at least one parse tree.

Summary

Computational recognition of human subtexts such as humor is a difficult task. Humor often depends on a change in context. The recognition of humorous intent depends on recognizing an intentional context change. Computational context understanding has not been achieved. To make previous context understanding easier, text can be semantically tagged.

Computational recognition of humorous intent can be divided into two parts: recognition of a humorous text, and recognition of the intent to be humorous.

A text is considered humorous if the first part of the text can have two meanings, or is compatible with two scripts or concepts. The meaning of the punchline script has to be incompatible with the default setup script.

The intent of the text is considered humorous if the schema related to the non-conflicting meaning of the setup was not the last schema that was activated in the preceding text.

The method could be initially tested on two-line question-answer dialogs. The first line of the dialog should contain a sound-alike word with a word in the second line or with a concept in the second line.

The success of the joke recognition and intent recognition will depend on the success of recognizing the appropriate wordplay.

Bibliography

S. Attardo, D. H. Attardo, P. Baltes, M. J. Petray (1994) "The linear organization of jokes: Analysis of two thousand texts," *Humor: International Journal of Humor Research*, 7:1.

D. Carroll (2004), *Psychology of Language*, Thompson Wadsworth, Belmont, California, 2004

M. Dascal (1985) "Linguistic use in jokes and dreams: Sociopragmatics vs. psychopragmatics," *Language and Communication*, 5(2)

D. Dooling & R. Lachman (1971), "Effects of Comprehension on Retention of Prose," *Journal of Experimental Psychology*

R. Hasan & M. Halliday (1976), *Cohesion in English*, Longman, London, 1986

R. Hetzron (1991) "On The Structure Of Punchlines," *HUMOR: International Journal of Humor Research,* 4:1

D. Jurafsky & J. Martin (2000), *Speech and Language Processing*, Prentice-Hall, New Jersey, 2000

D. Lenat (1995) "Steps to Sharing Knowledge." In N. Mars (Ed) *Toward Very Large Knowledge Bases*, IOS Press, 1995.

V. Raskin (1985), *The Semantic Mechanisms Of Humour*, Reidel, Dordrecht, 1985

G. Ritchie (2004), "Incongruity Resolution," Presentation at Forth International Summer School and Symposium on Humour and Laughter, Wolverhampton, England, 2004

J. Searle (1975), "Indirect Speech Acts," In P.Cole & J. Morgan (Eds) *Syntax and Semantics*: Vol. 3. Speech acts, New York, 1975