Pragmatically Computationally Difficult Pragmatics to Recognize Humour

Lawrence J. Mazlack
Applied Computational Intelligence Laboratory
University of Cincinnati
mazlack@uc.edu

Abstract
The humour found in short jokes and their often equivalent newspaper cartoons graphic representations are often dependent on the results of ambiguity in human speech. The ambiguities can be unexpected and funny. Sometimes well-known ambiguities cooperatively repeated can also be funny. Captioned cartoons often derive their humour from an unexpected ambiguity that can be understood by a listener who can automatically use world knowledge to resolve the ambiguity. The question considered here is whether the listener can be a computational device as well as a human and the pragmatic difficulty of applying linguistic pragmatics to do so. Computational analysis of natural language statements needs to successfully resolve ambiguous statements. Computerized understanding of dialogue must not only include syntactic and semantic analysis, but also pragmatic analysis. Pragmatics includes an understanding of the speaker’s intentions, the context of the utterance, and social implications of human communication, both polite and hostile. Computational techniques can use restricted world knowledge in resolving ambiguous language use. This paper considers the pragmatic difficulties in recognizing humour in short jokes as well as their representation in cartoons.

1. Introduction
Language does not come naturally to machines. Unlike humans, computers cannot easily distinguish between a river bank and a savings bank. Satire and jokes? Algorithms have great trouble with that. Irony? Wordplay? Cultural context and sensitivity? Maybe not. Whether computation can approach the human edge in decoding what things mean is an open question.

One liner jokes are similar to cartoons. Many comedians tell elaborate anecdotes. In contrast, some comedy routines consisting of telling simple one-liner jokes depict simple, cartoon-like situations, eliminating lengthy build-ups and going straight to the punch line. Sometimes, one-liner jokes are frequently repeated to the same listeners, with the listening audience expecting and desiring the joke.

While some cartoons have elaborate and lengthy build-ups and may depend on extensive world knowledge and/or experience (for example, the cartoon Dilbert that reflects a particular business working environment), most are simple, as are one-liner jokes. While there has been a fair amount of non-computational analysis of verbal humour and cartoons; there has only been a limited amount of computer based analysis of textually presented humour (Taylor 2008).

There has been even more limited computational analysis of cartoons; see Koponen (Koponen 2004) that reviews some non-computational cartoon analysis and provides a method of analysis that possibly could be computationally implemented. An important and interesting of the features of comics is the way that the written text and the pictures are combined. In comics, the two semiotic systems of word and image interact to create the full reading experience and often a humorous effect (Koponen 2004). The task of computationally analyzing the combination of text and images has not been achieved. This paper does not attempt to consider the interaction of text and images, but instead considers the pragmatic needs of short text analysis where the text has been extracted from cartoons.

2. The General Role Of Pragmatics
Within linguistic theory, the study of language use can be called pragmatics. The modern use of the term “pragmatics” can be traced back to Charles Morris (Morris 1938) who defined pragmatics as the study of “the relation of signs to interpreters”. He distinguished three fields of study: (1) syntax, the study of “the formal relations of signs to one another”, (2) semantics, the study of “the relations of signs to the objects to which the signs are applicable”, and (3) pragmatics, the study of “the relation of
signs to interpreters”. Signs fall into three classes: icons, indices, and symbols.

The meaning of indices can be found by describing rules for relating the sign to a context. These are pragmatic rules that are in essence action rules for finding relationships. The set of structures developed for these rules can be called pragmatic-semantic “trees” and divide into three categories:

* **Performatives** that describe the speaker’s intention or goal in using a sentence as a question, a command, etc.
* **Presuppositions**: Assumptions about the context that are necessary to make a sentence verifiable, or appropriate, or both.
* **Conversational Postulates**: A class of presuppositions concerning the dialogue that can be referred to as discourse codes of conduct (Bates 1976).

For semantic-pragmatic structures to operate, it is not enough to determine the meaning of individual words. Other types of information must also be used. In order to select between competing meanings, knowledge is required about the grammatical functions represented by the word orders in the natural sentence. Also, knowledge about the “real world” (linguistically known as presuppositions) is needed; i.e., the context in which the utterance took place. Practically or pragmatically speaking, computationally recognizing context from the text alone is difficult.

Along with contextual knowledge, a semantic-pragmatic structure needs to account for speech acts (performatives). Speech acts demonstrate the speaker’s goal. They can be a command, a question, a statement, etc. In other words, the associated meaning and the implied action must be understood. The theory must account for the fact that the listener or reader of the statement understands both (Bates 1976).

Next, the semantic-pragmatic structure must clarify the speaker’s ability to understand sequences of language which should mean one thing but clearly mean another (conversational postulates). It is assumed that normal human beings who enter into a conversation have agreed to be cooperative. Among several other things, this means: speakers will tell each other the truth, that they will only offer information thought to be new and relevant to the listener, and will only request information that they sincerely want.

This can be summarized and restated as: pragmatics can be characterized by the relationship between signs and their human users. Although not explicitly stated, most treatises on pragmatics focus on linguistic communication in context. Practically (or pragmatically), to fully analyze texts for possible humorous content, context pragmatics must be part of the analysis.

Precisely defining pragmatics is difficult. After a lengthy analysis, Levinson (Levinson 1983) sums up that despite problems the most promising definitions are those “that equate pragmatics with ‘meaning minus semantics’, or with a theory of language understanding that takes context into account, in order to complement the contribution that semantics makes to meaning”.

In summary, pragmatics is a subfield of linguistics that studies the ways in which context contributes to meaning. Pragmatics encompasses speech act theory, conversational implicature (i.e., what is suggested in an utterance (Grice 1975)), talk interaction, and other approaches to language behavior in philosophy, sociology, and linguistics (Mey 1993). It studies how the transmission of meaning depends not only on the linguistic knowledge (e.g. grammar, lexicon etc.) of the speaker and listener, but also on the context of the utterance, knowledge about the status of those involved, the inferred intent of the speaker, and so on. In this respect, pragmatics explains how language users are able to overcome apparent ambiguity, since meaning relies on the manner, place, time etc. of an utterance (Mey 1993). The ability to understand another speaker’s intended meaning is called pragmatic competence.

### 3. Supplying Pragmatics For Computational Analysis

There are multiple purposes in computationally attempting to understanding natural language; for example, understanding commands, query answering, and building models of customer’s desires. All of these have a history dating back to the 1960s. Generally, they are part of a cooperative dialogue. Question answering systems often make use of techniques for including pragmatic analysis in understanding natural language. Of interest to this paper are efforts that attempt to discriminate meaning (Beun and van Eijk 2004).

Potentially, analysis dependent on pragmatics can be used to analyze cartoons (Mazlack and Paz 1982). In this paper, dialogues extracted from cartoons will be analyzed to illustrate how pragmatic analysis could be used in considering cartoons.

#### 3.1 Cooperative Dialogue

Grice (Grice 1975) put forth the cooperative principle. The cooperative principle goes both ways: speakers (generally) observe the cooperative principle, and listeners (generally) assume that speakers are observing it. People who obey the cooperative principle in their language use will make sure that what they say in a conversation furthers the purpose of that conversation. Cooperative dialogue is often a computational goal. Computationally recognizing whether a dialogue is cooperative is difficult; however, whether or not a dialogue is cooperative is a necessary element in computationally recognizing humour.
Often, an accidentally non-cooperative dialogue is considered to be humorous. However, cooperative dialogues can also be humorous; for example: Knock-Knock jokes (Taylor and Mazlack 2004) or frequently expressed one-liner jokes that have become familiar and desired parts of standup comedian’s routines.

There have been many attempts at cooperative database query systems, some more successful than others. Research in cooperative responses has had a long history and has produced many useful algorithms and observations (Mays 1980) (Pollack, Hirschberg and Webber 1982) (Joshi, Webber and Wishedel 1986). However, it has had limited impact on the recent commercialization of dialogue technologies (Polifroni and Walker 2008) as it appears that: (1) there has been relatively little research showing that cooperative responses lead to more natural, effective, or efficient dialogues and (2) Previous work has hand-crafted responses or hand-annotated the database to support them that has made it difficult to port and scale then (Kaplan 1984) (Kalita, Colburn and McCalla 1984) (Cholvy 1990) (Polifroni, Chung and Seneff 2003) (Benamara 2004).

One barrier to the acceptance of natural language database query systems is the substantial installation effort required for each new database. Much of this effort involves the encoding of semantic knowledge for the domain of discourse, necessary to correctly interpret and respond to natural language questions. For such systems to be practical, techniques must be established to increase their portability to new domains. A typical early cooperative database query system is CO-OP (Kaplan 1984). CO-OP derives its domain-specific knowledge from a lexicon (the list of words known to the system) and the information already present in the structure and content of the underlying database. CO-OP determines from a question what information is required, i.e., the direct, literal, and correct response, as well as the information that the questioner is unaware of important pertinent facts not explicitly requested in the question. For example:

> Human Question: How many students failed CSE in Spring 2012?
< Cooperative System answer: None, CSE110 was not given in Spring 2012
< Non-Cooperative answer: The answer of “zero” would not be cooperative although it would be accurate.

The user who posed the above question presumed that the CSE110 class was taught in the Spring of 2012. A cooperative system, on finding that this presumption was false, responds with a corrective indirect response by supplying the negated presumption with a change of focus in the original question.

Non-cooperative dialogue in cartoons often leads to funny results when their statements are ambiguous. In a Tiger cartoon (Figure 1), there is the dialogue:

> Boy1: Wow! What a chase! Our dog Stripe and Mrs. Parker’s cat!
< Boy2: Nope, Stripe got away.

An example of a cooperative response is in the answer: “Did he catch him? ”Prior to this question, Boy1 in the dialogue only knows that there is a chase going on and there are two participants: Stripe and Mrs. Parker’s cat. The situation is ambiguous because we do not know who is chasing whom. Here, the common human presumption is that dogs (Stripe) usually chase cats (Mrs. Parker’s cat). The computer on finding this presumption to be false, could respond with a corrective indirect response: “Nope, Stripe got away” rather than with the direct answer of “no”. In order for the computer to recognize that this is a joke, it must somehow have the pragmatic world knowledge that dogs usually chase cats and that the dialogue is humorous.

Another type of cooperative response is when the system replies with a suggestive indirect response. It is appropriate for a cooperative answer to contain relevant information to be requested in a follow-up situation. A cooperative heuristic is to change the focus of the original question. The focus is that aspect of the question that is most likely to shift in a follow-up question.

In the dialogue drawn from a B.C. cartoon (Figure 2),

> Caveman1: Dog census.
> Caveman1: Do you have a dog?
< Caveman2: Yes.
> Caveman1: Name please ...
< Caveman2: Fido.
> Caveman1: And your dog’s name, Mr. Fido?

There are several things going on in this dialogue. First, there is the background knowledge that Fido is always a dog’s name; how this background knowledge can be computationally supplied/understood is a matter of continuing speculation and is practically (pragmatically) difficult. Another thing going on is that the dialogue could have used a suggestive indirect response. In this example, a census taker (Caveman1) is asking questions of a subject (Caveman 2) that is responding as if it is a naïve computer. A common presumption here is that the first question that a census taker will ask is the subject’s name. (Presumptions can also be discussed in terms of frames; frames will be approached in a subsequent section of this paper.) The naïve respondent needs to recognize the ambiguous situation created by the question: “Name, please”. If a potential change of focus is anticipated, the question could be more unambiguously seen as two more cooperative questions:
3.2 Restricted Discourse Domain

Another early database system (Harris 1978), uses the database itself to find the use of words in the question, to build expectations, and to resolve ambiguities. The system interprets input based on what makes sense with its limited world model, the database. The goal is to eliminate the creation of a semantic store purely for the natural language parser, since a different structure would be required for each domain of discourse. Instead the database itself is used as the primary knowledge structure within the system. When successful, the parser can be interfaced to other databases with only minimal effort.

In processing ambiguous statements, several interpretations may arise. A heuristic used is unintentional interpretations of input questions are usually not false for the specific domain, but have a vacuous response (Coles 1972). To use this heuristic, these interpretations are posed to the database as queries. If all interpretations fail to find a response, then the answer is: “There aren’t any”. This negative answer assumes that the dialogue will only be about information contained in the database. If more than one interpretation can be answered successfully, then the system enters into a clarification dialogue, similar to what humans would have to do when faced with an ambiguous question. If exactly one interpretation is found, then the system responds using this interpretation for the question.

In a cartoon Garfield (Figure 3), the human speaker asks the cat Garfield to play with Nermal (the kitten). As a result, Garfield plays with Nermal as if Nermal is a toy ball (Garfield bounces Nermal up and down). The following ambiguous situation is created:

- “Play with Nermal” means that Nermal is a toy.
- or
- “Play with Nermal” means that Nermal and Garfield should play together as friends.

The application of pragmatic world knowledge would eliminate the first choice. However, it is not clear how a computational device would discern what should be done. Perhaps, the computer could search the toy and friend domains for an entry Nermal. On not finding Nermal in the toy domain, and, finding Nermal in the friend domain, the ambiguous situation would be resolved. However, this is not a simple, easy solution and might entail developing fresh domains for each group of situations.

In the Wizard Of Id cartoon’s dialogue (Figure 4):

- Knight: Our records show that there is a dip in unemployment.
- King: Find him and fire him.

the Knight is saying to the King that “our records show that there is a dip in unemployment” and is perhaps implying the question “What do we do next?” The ambiguity comes from the fact that “dip” can have many different meanings, including either a foolish person or a downward trend. To resolve the situation, the Knight could have added the information:

- Knight: Dip in unemployment is a paraphrase of: temporary decline in the unemployment statistics.

3.3 World Knowledge Using Frames And Scripts

Many computational mechanisms provide world knowledge to a situation description, or a dialogue, using frames or scripts. Frames, simply put, are highly structured sets for keeping pragmatics, or, stereo typical situations (Minsky 1975). When an action is carried out, some canonical description is stored in the frame that will permit the program to reconstruct the context in which in which the event took place. Frames include syntactic information such as: subject, object, prepositional phrase, as well as: semantic and pragmatic facts that provide various reasons, situations, and purposes not explicitly stated. Frames carry over to the subsequent statements and conventions that mark anaphora or presupposition link the program to slots in the current or past program to slots in the current or past frames that will resolve the reference. As an example, Minsky offered a child’s birthday party as the kind of thing that a frame might represent. A birthday party frame includes slots, such as food, games, and presents, that specify general features of the events. Slots are bound to fillers, or representatives of the particulars of the situation.

Scripts are like frames in that they also have empty slots that are filled with the context from a dialogue, text, or sequence of actions. Scripts can provide world knowledge about common experiences or situations in terms of Schank’s (Schank 1972) conceptual dependency primitives. A text is understood by mapping sentences into actions or primitive acts as described in the script. Unstated facts described in a script but not in the sentence are supposed to be true. This provides a “background” or world knowledge for understanding and reasoning (Shank 1975). Schank’s best known script example is a restaurant script that takes someone through the sequence and mechanics of eating at a restaurant (Schank and Abelso 1977).

Humour can occur when a frame shift occurs within a text. The frame shift is usually unexpected by the listener,
but can be familiar, even desired or expected. For example, the well known standup comedian Henny Youngman’s “Take my wife, please!” can be prefaced by a short statement but the punch line is expected and desired:

> The ladies really like to shop, now ...
> Take my wife, please!

This is an example of a script shift as the expected second part of the anecdote would be something about a shopping experience. (Youngman explained the origin of his classic line “Take my wife, please” as a misinterpretation: He took his wife to a show and asked the usher to escort his wife to a seat. But his request was taken as a joke, and Youngman used the line countless times ever after.) Another, similar, familiar short script violation is:

> That was no lady, it is my wife

Humour can be said to be the result of a semantic leap or frame shift (Coulson 2001). However, the next frame element must conceptually blend with the first part or it is likely that the listener, with unlinked parts, would consider the result to be not funny. Sometimes, understanding the semantic leap is helped by recognize the humorous intention of a dialogue; however, recognizing intention is a pragmatically difficult task (Taylor and Mazlack 2005).

A Beetle cartoon can be used to illustrate the frame and the script concepts.

> Captain: How’s the machine gun practice, Zero?
< Zero: Fine sir, I hit the target every time!
> Captain: Good! Now try shooting that big gun over by Sarge.
< Zero: Right (shoots the machine gun at another gun that Sarge is standing next to.

Here, the relative pronoun must be resolved in the phrase “that gun”. Two possibilities are:

> Use the machine gun to shoot at the gun next to Sarge; i.e., use that gun as a target.

or

> Use the gun next to Sarge to shoot with and shoot at the original target.

The ambiguous situation can be resolved by using world knowledge about target practice. For example, it is helpful to know that targets should not be expensive, useful things, i.e., a gun; and, that a target is not located near a human.

This type of information can be provided by demons (Bratko 2011) in the case of frames or by the reason or goals statement in the case of scripts. (A demon is a facet of a slot in a frame that causes some action to be taken when the frame is accessed in certain types of ways. For example, an if-needed demon is activated or triggered if the value of the slot is required and a value has not yet been stored in the slot, and it should calculate or otherwise obtain a value for the slot, while a range demon is triggered if a new value is added to the slot, to check that the value added is permissible for this particular slot.)

Frames can also capture humour where the joke does not come from an overlapping frame shift or surprise, but instead from conflicting world views in a cooperative dialogue as in the Arlo & Janis cartoon in Figure 6. Not everyone finds every cartoon funny. For example, the situation shown in Figure 6 may be funnier to men as opposed to women as the desire to possibly have more children may be culturally more important to women than to men.

4. Summary

The humour found in short jokes and their often equivalent newspaper cartoons graphic representations are often dependent on the results of ambiguity in human speech. Often, the humour comes because the ambiguities are unexpected. Sometimes well-known ambiguities cooperatively repeated can also be funny.

One liner jokes and many cartoons are similar in comedic effect. They depend on the type of comedy that Freud described as compact and condensed, with double meanings; this is to fool a person’s simple minded internal censors that initially see only innocent surface meanings (Freud 1905).

Captioned cartoons often derive their humour from an unexpected ambiguity that can be understood by a listener who can automatically use world knowledge to resolve the ambiguity. The question considered in this paper is whether the listener can be a computational device as well as a human and the necessary or pragmatic difficulty of applying linguistic pragmatics.

An important and interesting feature of comics is the way that the written text and the pictures are combined. In comics, the two semiotic systems of word and image interact to create the full reading experience and often an enhanced humorous effect. In some cartoons, the images are not critical to the joke understanding and the text can be extracted for easy analysis. In others, the pictures aid in quickly understanding the joke, the images help in Figures 3 and 5 while they are not needed in the other figures.

Computational analysis of natural language statements needs to successfully resolve ambiguous statements. Computerized understanding of dialogue must not only include syntactic and semantic analysis, but also pragmatic analysis. Pragmatics includes an understanding of the speaker’s intentions, the context of the utterance, and social implications of human communication, both polite and hostile. Computer techniques have been developed that use restricted world knowledge in resolving ambiguous language use.

Humour can be the result of a semantic leap or frame shift. However, the new frame element must conceptually blend with the first part of the text or it is likely that the
listener would not consider the result to be funny. Sometimes, understanding the semantic leap is helped by recognize the humorous intention of a dialogue; however, recognizing intention is a pragmatically difficult task.

This paper considered the practical (pragmatic) difficulties in recognizing humour in short jokes as well as their representation in cartoons. Applying linguistic pragmatics was considered; doing this is probably beyond current computational capabilities.

References


