Computational Humor

Kim Binsted, *University of Hawaii*

No, this is no April Fool's prank. Computer scientists at labs around the world are conducting serious research into … humor. Although it might seem whimsical, many excellent reasons exist to take a closer look at this fascinating aspect of human cognition and interaction.

Humor affects attention and memory, facilitates social interaction, and ameliorates communication problems. If computers are ever going to communicate naturally and effectively with humans, they must be able to use humor. Moreover, humor provides insight into how humans process language—real, complex, creative language, not just a tractable subset of standard sentences. By modeling humor generation and understanding on computers, we can gain a better picture of how the human brain handles not just humor but language and cognition in general.

In popular perception, computers will never be able to use or appreciate humor. Fictional computers and robots are almost always portrayed as humorless, even if they're skilled at natural language, graceful bipedal motion, and other human behaviors that current machines find challenging. Then again, chess too was once thought to be the sole domain of humans, and now computer programs play at the grandmaster level.

These four articles focus on different aspects and applications of humor. Benjamin Bergen and Seana Coulson propose a model of humor comprehension based on frame-shifting within a simulation-based natural-language-understanding system. Anton Nijholt describes how embodied agents can use humor to make human-agent interaction more effective and believable. Oliviero Stock and Carlo Strapparava describe humor's effects on attention and memory and describe an application of computational humor to advertising. Finally, Graeme Ritchie, Ruli Manurung, Helen Pain, Annalu Waller, and Dave O'Mara describe an interactive riddle builder called Standup, which helps children with communication-related disabilities use humor to interact with other children. One day, perhaps, thanks to the groundbreaking research these articles describe, computers will be able to devise their own April Fool's pranks.

—Kim Binsted

Frame-Shifting Humor in Simulation-Based Language Understanding

Benjamin Bergen, *University of Hawaii, Manoa*
Seana Coulson, *University of California, San Diego*

In an effort to focus on tractable problems, computational natural-language-understanding systems have typically addressed language phenomena that are amenable to combinatorial approaches using static and stereotypical semantic representations. Although such approaches are adequate for much of language, they're not easily extended to capture humans' more creative language interpretation capacities. An alternative tack is to begin by modeling less typical, more complex phenomena, with the goal of encompassing standard language as a trivial case.[1]

Semantic interpretation

Suppose you hear someone saying, "Everyone had so much fun diving from the tree into the swimming pool, we decided to put in a little … " At the point in the sentence where you hear the words "put in," you've already committed to an interpretation of the clause—probably that the owners are installing a diving board. Indeed, psycholinguistic research suggests that human sentence processing is both incremental and predictive, because people integrate perceptual input with linguistic and conceptual information at multiple levels of representation.[2] Neuroimaging research with magnetoencephalography suggests that after an initial modality-specific processing stage of approximately 200 milliseconds, speech processing is subserved by a bilateral network of inferior prefrontal and temporal lobe regions that are simultaneously active for hundreds of milliseconds.[3] Such data suggest that the processes of lexical access, semantic association, and contextual integration are simultaneous, as represented in cascade models.

Besides its empirical motivation, incremental semantic processing has computational benefits for word recognition and contextual integration. Word recognition in natural speech, for example, is extremely challenging because of extensive variability in the acoustic input. Top-down semantic information greatly facilitates segmentation of the sound stream. Moreover, spoken language is produced quickly—typically at about two to three words per second—which necessitates parallel processing of linguistic cues and their semantic referents. Besides easing word recognition, activating the correct frame of reference greatly facilitates contextual integration. For instance, in the swimming pool example, knowledge of diving and the typical backyard pool lets you more easily recognize and inte-

1541-1672/06/$20.00 © 2006 IEEE
Published by the IEEE Computer Society

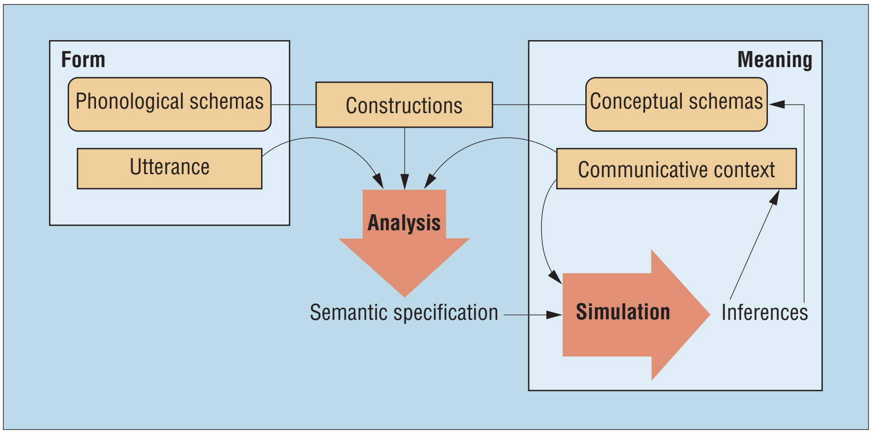


Figure 1. Overview of the Embodied Construction Grammar architecture with two core processes, analysis and simulation.[5]

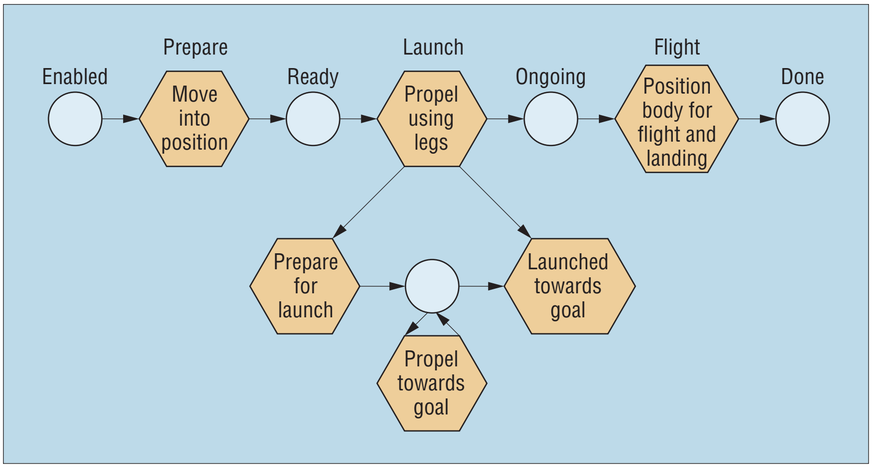


Figure 2. A simplified Dive X-schema.

grate expected items, such as a diving board, into the scenario.

However, one thing that differentiates human sentence processing from many computational systems is humans' capacity to deal with unexpected input, even when it necessitates revising information that they've already assembled. People, for example, are readily able to interpret the sentence "Everyone had so much fun diving from the tree into the swimming pool, we decided to put in a little *water*." The word "water" prompts you to revise the default assumption that the pool had water in it. Revising this simple assumption has substantial implications for the consequences of diving from the tree into the pool, and for the mindset of those who enjoy such activities. This reanalysis process, known as *frame-shifting*, highlights the need for dynamic inferencing in language interpretation and has been argued to be a test case for models of meaning construction.[1]

Natural-language-understanding systems might achieve the flexible, interpretive capacity necessary for frame-shifting by adopting an empirically inspired architecture that is based on dynamic internal imagery. In such models, language interpretation involves creating an internal simulation of events that includes the sensory, motor, and affective dimensions. Research suggests that the neural systems responsible for performing actions or perceiving percepts are also recruited for linguistically inspired simulations. In other words, like dreaming, recalling, and imagining, language processing uses human perceptual and motor systems as internal models that allow for the construction of subjective experiences in the absence of motor action or perceptual input.[4] Because our experiences with pools almost without exception include water, water will automatically be activated in mental simulations that involve pools. Our experience with diving, by contrast, presumably involves some cases of landing on a solid surface, thus enabling us to viscerally imagine diving into a pool with no water.

Simulation-based natural-language-understanding systems

One natural-language-understanding system that employs a simulation-based architecture was developed under the rubric of Embodied Construction Grammar.[5] The ECG language-understanding system has two main processes: *analysis* and *simulation* (see figure 1). The analysis system parses each input utterance into constituent words and syntactic structures, using a representation of the current communicative context as well as stored pairings of phonological and conceptual knowledge known as *constructions*. The simulation system runs dynamic simulations of those utterances' content, producing inferences and updating beliefs about the world.

The ECG simulation system uses a representational formalism known as the *X-schema*. X-schemas are based on stochastic Petri nets and have numerous desirable properties. They are dynamic and probabilistic and allow for both parallel processing and hierarchical structure. Figure 2 shows a simplified representation of the Dive X-schema—that is, the machinery used to virtually perform or mentally simulate diving. Any individual diving event, as with any complex action, will be a particular instantiation of this general X-schema. In the case of diving, instances will differ in the agent who performs the dive, the dive's target, the force exerted, the trajectory, and so on. The language system must supply these parameters of the Dive X-schema, but it can use default values in their absence. The simulation system depends on the analysis system to provide it with information about which X-schemas to run, what parameterizations to assign to them, and how to bind them together.

The analysis system uses linguistic input to produce a specification of the mechanisms the simulation system uses. This parameterized interface is known as the *semantic specification* (see figure 1). The inputs to the analysis system are contextually activated schemas, along with the words and other constructions the utterance activates; constructions map aspects of form to aspects of meaning. As we noted earlier, the output of analysis is the semantic specification. For example, the word "diving" maps from a sound or character sequence to the X-schema representation for diving (see figure 2), just as the word "tree" will map its sound or spelling to a parameterization of how to simulate a tree. Larger grammatical constructions indicate how these schematic, parameterized representations of simulatable meaning are bound together. Consequently, an English speaker knows that the subject of the verb "dive" will be the agent performing the dive, and that the subsequent prepositional phrases describe the locus of its starting and ending points.

The processes in the ECG system are interdependent: the simulation system depends on input from the analysis system, and defective simulation can force a novel analysis. Moreover, the processes proceed incrementally. Critically, the simulation system engages as early as possible and need not wait for the analysis process to terminate. Indeed, as we argued earlier, simulation must begin as soon as a coherent chunk of a semantic specification has been assembled.

Computational-humor processing

Humans naturally and frequently produce humorous language. So, to interact as a human-like conversational agent or to serve as a scientific model of human-language understanding, a natural-language-understanding system must be able to deal appropriately with linguistic humor. Early simulation is a key feature that facilitates humor processing in computational natural-language-understanding systems. Here's an example of how the ECG system deals with an utterance such as "Everyone had so much fun diving from the tree into the swimming pool, we decided to put in a little water."

A processing example

First, assuming for simplification that the input is in text form rather than spoken, analysis begins from left to right (starting with the first word) and hypothesizes lexical and larger linguistic constructions that could account for the input data, using a chart-parsing algorithm.[6] As the system reads in each new word, the analysis process maintains numerous competing candidate sets of lexical and grammatical constructions. Each coherent candidate set includes not only these constructions, but also bindings among them. For instance, the example sentence activates lexical constructions representing words—"everyone," "had," "so," and so on—as well as larger grammatical constructions such as noun phrases ("the swimming pool") and prepositional phrases ("from the tree").

The semantic side of these assembled networks of constructions contains precisely the parameterized schematic representations the simulation device needs. Once a single analysis surpasses a defined certainty threshold, the analysis process reads its semantic specification into the simulation engine. So, after the first clause, the analysis process passes the single best analysis into simulation. The simulator runs an internal simulation of this experience, activating the appropriate X-schema with the parameterizations determined through analysis, such as who was performing the action, with what trajectory, and so on. The system fills in those aspects of simulation not specified from the language with default or contextually inferable values, so it represents the pool as filled with water. The simulation generates appropriate affective and encyclopedic inferences so that it can update its beliefs about the world—people in the scene were using the swimming pool in its normal function, that is, diving in, getting wet, and enjoying it.

> To interact as a human-like conversational agent, a natural-language-understanding system must be able to deal appropriately with linguistic humor.

As processing continues, however, the system is confronted with a semantic incoherency. Namely, when it encounters the word "water" at the end of the sentence and simulates the second clause's content, the result is inconsistent with the previous simulation. The highest-level construction in the sentence, the "X is so Y that Z" construction, indicates that the events the two clauses describe (the enjoyment of the diving and the putting in of water) are related. However, unless heavy contextual biasing exists (for instance, if the system previously knows the pool to be full of jello), the simulation system can't coherently maintain the simulation it constructed for the first clause as a predecessor of the simulation for the second clause; without water in the pool, people can't have been diving normally into it. Because the simulation system can't establish a causal or temporal relationship between the two partial simulations it performed for the two clauses, it fails and calls on the analysis system to provide a new semantic specification. The success of a semantic specification that calls for a different simulation in the first clause—namely, a slightly different X-schema for the first event—results in a coherent relationship between the two clauses' simulations.

Our system's strengths

Although this example is a mere sketch of the processing that occurs in a simulation-based natural-language-understanding system, it illustrates several important features of this approach. First, using detailed semantic interpretation in the form of simulation is vital to correctly identifying semantic incoherencies, because nothing about the humorous utterance's linguistic form itself—the words it uses or their formal combination—is conceptually discrepant in any way. Second, the early (utterance internal) commitment to semantic interpretation predicts differences in the processing of humorous utterances that require frame-shifting from those that are merely ambiguous. For instance, in processing the sentence "That book sounds so great I'm going to dive right into it," the word "dive" has a different sense than when it describes physical diving. However, in this case, prior linguistic context (and simulation) biases the processor toward the correct interpretation. Third, a simulation-based model uses real-world knowledge to capture subtle semantic differences in the meanings of particular words that arise in the simulation and not in the semantic specification. For example, only the simulation can distinguish between the diving you would do into a pool and on a football field, or between diving into a pool with versus without water. Finally, the process of misunderstanding and correction—committing to a simulation only to be confronted with an incoherency that needs correcting—drives the humorous effect the utterance has on humans. A simulation-based architecture exhibits similar behavior.

Conclusions

As natural-language-understanding systems approach human-like conversational competence, a main challenge is dealing with language that requires deep conceptual knowledge about the details of human experience, as humorous language often does. Incorporating simulation devices into such applications can begin to solve the problem of how to use knowledge of the world to improve human-machine communication. Computational tools developed to deal with the complexities of linguistic humor might also apply to other challenging problems in natural language interpretation.

References

1. S. Coulson, *Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction*, Cambridge Univ. Press, 2001.
2. Y. Kamide, G.T.M. Altmann, and S.L. Haywood, "The Timecourse of Prediction in Incremental Sentence Processing: Evidence from Anticipatory Eye Movements," *J. Memory and Language*, vol. 49, no. 1, 2003, pp. 133–156.
3. K. Marinkovic, "Spatiotemporal Dynamics of Word Processing in the Human Cortex," *Neuroscientist*, vol. 10, no. 2, 2004, pp. 142–152.
4. L.W. Barsalou, "Perceptual Symbol Systems," *Behavioral and Brain Sciences*, vol. 22, no. 4, 1999, pp. 577–609.
5. B. Bergen and N. Chang, "Embodied Construction Grammar in Simulation-Based Language Understanding," *Construction Grammars: Cognitive Grounding and Theoretical Extensions*, J. Ostman and M. Fried, eds., John Benjamins, 2005, pp. 147–190.
6. J. Bryant, "Constructional Analysis," master's thesis, Computer Science Dept., Univ. of California, Berkeley, 2004.

Embodied Conversational Agents: "A Little Humor Too"

Anton Nijholt, *University of Twente*

Social and intelligent agents have become a leading paradigm for describing and solving problems in human-like ways. In situations where it's useful to design direct communication between agents and their human partners, the display of social and rational intelligence in an embodied human-like agent (that is, an agent visualized onscreen as an animated character) allows natural interaction between the human and the agent that represents the system the human is communicating with. Research in intelligent agents includes reasoning about beliefs, desires, and intentions. Apart from contextual constraints that guide the agent's reasoning and behavior, other behavioral constraints exist that follow from models that describe emotions. These models assume that emotions emerge based on appraisals of events taking place in the environment and how these events affect goals that the agents are pursuing. In current research, it's also not unusual to incorporate personality models in agents to adapt this appraisal process as well as reasoning, behavior, and display of emotions to personality characteristics. So, we can model a lot of useful and human-like properties in artificial agents, but, in Roddy Cowie's words, "If they are going to show emotion, we surely hope that they would show a little humor too."[1]

> Humor helps to regulate a conversation and can help to establish common ground between conversational partners. It makes conversation enjoyable and supports interpersonal attraction.

What are the benefits of using humor? Humor helps to regulate a conversation and can help to establish common ground between conversational partners. It makes conversation enjoyable and supports interpersonal attraction.[2] According to Deborah Tannen, humor makes your presence felt.[3] Many researchers have also mentioned its benefits in teaching and learning and have made this role explicit in experiments. Humor contributes to motivation, attention, comprehension and retention of information, and the development of affective feelings toward content. It helps create a more pleasurable learning experience, fosters creative thinking, reduces anxiety, and so on.

Embodied agents using humor

Embodied conversational agents (ECAs) have been introduced to play the role, among others, of conversational partner for the computer user. Rather than addressing the machine, the user addresses virtual agents that have particular capabilities and are responsible for certain tasks. The user might interact with ECAs to engage in an information service dialogue or a transaction, to solve a problem cooperatively, to perform a task, or to engage in a virtual meeting. In these kinds of situations, humans use humor to ease communication problems. In a similar way, humor can help solve communication problems that arise within human-ECA interaction.

Researchers working on the *Computers Are Social Actors* paradigm[4] have convincingly demonstrated that people interact with computers as if they were social actors. Depending on the way we can program a computer to interact, people might find it polite, dominant, extroverted, introverted, or any other attitudes or personality traits we can think of. Moreover, people react to these attitudes and traits as if a human being were displaying them. From the CASA experiments, we can extrapolate that humor, because of its role in human-human interaction, can play an important role in human-computer interaction. Experiments examining humor's effects in task-oriented computer-mediated communication and in human-computer interaction have confirmed this.

More and more, we see ECAs employed in 2D or 3D virtual-reality educational and entertainment environments, e-commerce applications, and training and simulation environments.[5] Research projects suggest that in the near feature, we might expect that in addition to being domain and environment experts, ECAs will act as personal assistants, coaches, and buddies. They will accompany their human partners, migrating from displays on handheld devices to displays embedded in ambient-intelligence environments. Natural interaction with these ECAs will require them to display rational and social intelligence and, indeed, also a little humor when appropriate and enjoyable.

Computational humor

Well-known philosophers and psychologists have contributed their viewpoints to the theory of humor. Sigmund Freud saw humor as a release of tension and psychic energy, while Thomas Hobbes saw it as a means to emphasize superiority in human competition. In the writings of Immanuel Kant, Arthur Schopenhauer, and Henri Bergson, we can see the first attempts to characterize humor as dealing with incongruity—that is, recognizing and resolving incongruity. Researchers including Arthur Koestler, Marvin Minsky, and Alan Paulos have tried to clarify these notions, and Victor Raskin and Graeme Ritchie have tried to formally describe them.

As you might expect, researchers have taken only modest steps toward a formal theory of humor understanding. General humor understanding and the closely related area of natural-language understanding require an understanding of rational and social intelligence, so we won't be able to solve these problems until we've solved all AI problems. It might nevertheless be beneficial to look at the development of humor theory and possible applications that don't require a general theory of humor; this might be the only way to bring the field forward. That is, I expect progress to come from application areas—particularly in games and other forms of entertainment—that require natural interaction between agents and their human partners, rather than from investigations by a few researchers into full-fledged theories of computational humor.

Incongruity-resolution theory provides some guidelines for computational-humor applications. I won't look at the many variants that have been introduced or at details of one particular approach. Generally, I follow Graeme Ritchie's theory;[6] however, since I prefer to look at humorous remarks that are part of the natural interaction between an ECA and its human conversational partner, my starting point isn't joke telling or pun making. Rather, I assume a small piece of discourse consisting of two parts. You read or hear and interpret the first part, but as you read or hear the second part, it turns out that a misunderstanding has occurred that requires a new, probably less obvious interpretation of the text. So, we have an obvious interpretation, a conflict, and a second, compatible interpretation that resolves the conflict. Although misunderstandings can be humorous, this isn't necessarily the case. Deliberate misunderstanding sometimes occurs to create a humorous remark, and it's also possible to construct a piece of discourse so that it deliberately leads to a humorous misunderstanding. In both cases, we need additional criteria to decide whether the misunderstanding is humorous. Criteria that humor researchers have mentioned deal with a marked contrast between the obvious interpretation and the forced reinterpretation, and with the reinterpretation's commonsense inappropriateness. As an example, consider the following dialogue in a clothing store:

Lady: "May I try on that dress in the window?"

Clerk (doubtfully): "Don't you think it would be better to use the dressing room?"

The first utterance has an obvious interpretation. The clerk's remark is confusing at first, but looking again at the lady's utterance makes it clear that a second interpretation (requiring a different prepositional attachment) is possible. This interpretation is certainly different, and, most of all, it describes a situation that you might consider inappropriate.

> General humor understanding requires an understanding of rational and social intelligence, so we won't be able to solve this problem until we've solved all AI problems.

What can we formalize here, and what formalisms are already available? AI researchers have introduced scripts and frames to represent meanings of text fragments. In early humor theory as it relates to AI, these knowledge representation formalisms were used to intuitively discuss an obvious and a less-obvious (or hidden) meaning of a text. A misunderstanding allows at least two frame or script descriptions of the same piece of text; the two scripts involved overlap. To make it clear that the nonobvious interpretation is humorous, at least some contrast or opposition between the two interpretations should exist. Script overlap and script opposition are reasonably well-understood issues, but until now, although often described more generally, the attempts to formalize this opposition mainly look at word-level oppositions (for example, antonyms such as hot versus cold). Inappropriateness hasn't been formalized at all.

Conversational humor: Constructing humorous acts

People smile and laugh when someone uses humor. It's not necessarily because someone pursues the goal of being funny or of telling a joke, but because the conversational partners recognize the possibility of making a funny remark—deliberately, spontaneously, or something in between, taking into account social display rules. It's possible to look at some relatively simple situations that let us make humorous remarks. These situations fit in the explanations I gave earlier, and they make it possible to zoom in on the main problems of humor understanding: rules to resolve incongruity and criteria that help determine whether a solution is humorous.

Here I talk about surprise disambiguations. We can have ambiguities at pragmatic, semantic, and syntactic levels of discourse (text, paragraphs, and sentences). At the sentence level, we can have ambiguities in phrases (for example, prepositional-phrase attachment), words, anaphora, and, in the case of spoken text or dialogue, intonation. As we interpret text that we read or hear, misunderstandings will become clear and be resolved, maybe with help from our conversational partner. Earlier, I gave an example of ambiguity that occurred because a prepositional phrase could be attached to a syntactic construct (a verb, a noun phrase) in more than one way. My main line of research, however, deals with ambiguities in anaphora resolution. For example, consider the following dialogue:

Adviser: "Our lawyers put your money in little bags, then we have trained dogs bury them around town."

Dilbert: "Do they bury the bags or the lawyers?"

Here, "them" is an anaphor referring to a previous noun phrase. You can find antecedents among the noun phrases in the first sentence.

From a research viewpoint, the advantage of looking at such a simple, straightforward humorous remark is that we can confine ourselves to just one sentence. So, rather than having to look at scripts, frames, and other discourse representations, we can concentrate on the syntactic and semantic analysis of just one sentence. For this analysis, I use well-known algorithms that transform sentences into feature structure representations and issues such as script overlap and script opposition into properties of feature sets. Moreover, many algorithms for anaphora resolution are publicly available. Erroneous anaphora resolution with the aim of creating a humorous remark can make use of properties of possible anaphora antecedents.

Contrast and inappropriateness are global terms from (not yet formalized) humor theory. In my approach, determining contrast translates into a heuristic that considers a potentially humorous antecedent and decides to use it because it has many properties in common with the correct antecedent. However, at least one salient property distinguishes the two potential antecedents (a shop window versus a dressing room, a bicycle versus a car, a bag versus a lawyer). My approach checks for inappropriateness by looking at constraints associated with the verb's thematic roles in the sentence. For example, these constraints distinguish between animate and inanimate; hence, burying lawyers who are alive is inappropriate. Obviously, you can say more about this—lawyers are easier targets for jokes than pharmacists, for instance—but such observations are beyond this essay's scope.

Experiments and implementation

My research group is creating a chat bot that implements my approach to humorous anaphora resolution. One reason we chose a chat bot is that its main task is to get a conversation going. Hence, it might miss opportunities to make humorous remarks, and when an intended humorous remark turns out to be misplaced, this isn't necessarily a problem. Implemented algorithms for anaphora resolution are available. We chose a Java implementation (JavaRAP) of Shalom Lappin and Herbert Leass' well-known Resolution of Anaphora Procedure.[7] We obtained a more efficient implementation by replacing the embedded natural-language parser with a parser from Stanford University. Currently, we and other researchers are designing experiments to find ways to deal with anaphora resolution algorithms' low success rate and to consider the introduction of a reliability measure before proceeding with possible antecedents of an anaphor. Other issues we're investigating are the different frequencies and types of anaphora in written and spoken text. In particular, we've been looking at properties from the anaphora viewpoint of conversations with well-known chat bots such as ALICE, Toni, Eugine, and Jabberwacky. Resources that we've investigated are publicly available knowledge bases such as WordNet, WordNet Domains, FrameNet, VerbNet, and ConceptNet. For example, in VerbNet, every sense of a verb is mapped to a verb class representing the conceptual meaning of this sense. Every class contains both information on the thematic roles associated with the verb class and frame descriptions describing how the verb can be used. This lets you check whether a possible humorous antecedent of an anaphor sufficiently opposes the correct antecedent because of constraints that it violates. Unfortunately, because VerbNet only contains about 4,500 verbs, many sentences can't be analyzed.

> It might be beneficial to look at the development of humor theory and possible applications that don't require a general theory of humor; this might be the only way to bring the field forward.

Conclusions and future research

As I mentioned earlier, when looking at the fundamental problems in humor research, we must wait until the main problems in AI have been solved and then apply the results to humor understanding. It's more fruitful to investigate humor itself and see whether solutions that are far from complete and perfect can nevertheless find useful applications. In games and entertainment computing, natural interaction with ECAs requires humor modeling. Although many forms of humor don't fit into my framework of humorous misunderstandings, I think it's a useful approach.

Current humor research has many shortcomings, which are also present in my approach. In particular, the conditions I've mentioned (such as contrast and inappropriateness) might be necessary, but they're far from sufficient. Further pinpointing of humor criteria is necessary. My approach lets me apply well-known theories from computational linguistics rather than obliging me to continue with archaic approaches and concepts. Moreover, my approach makes the issue of humor versus nonhumor much more visible than in older approaches.

References

1. R. Cowie, "Describing the Emotional States Expressed in Speech," *Proc. Int'l Speech Communication Assoc. Workshop Speech and Emotion*, Queens Univ., 2000, pp. 11–18; www.qub.ac.uk/en/isca/proceedings/pdfs/cowie.pdf.
2. A. Cann, L.G. Calhoun, and J.S. Banks, "On the Role of Humor Appreciation in Interpersonal Attraction: It's No Joking Matter," *Humor: Int'l J. Humor Research*, vol. 10, no. 1, 1997, pp. 77–89.
3. D. Tannen, *Conversational Style: Analyzing Talk among Friends*, Ablex Publishing, 1984.
4. B. Reeves and C. Nass, *The Media Equation*, Cambridge Univ. Press, 1996.
5. H. Prendinger and M. Ishizuka, eds., *Life-Like Characters: Tools, Affective Functions, and Applications*, Springer, 2004.
6. G. Ritchie, "Developing the Incongruity Resolution Theory," *Proc. AISB Symp. Artificial Intelligence and Creative Language: Stories and Humor,* Univ. of Sussex, 1999, pp. 69–75.
7. S. Lappin and H.J. Leass, "An Algorithm for Pronominal Anaphora Resolution," *Computational Linguistics*, vol. 20, no. 4, 1994, pp. 535–156.

Automatic Production of Humorous Expressions for Catching the Attention and Remembering

Oliviero Stock and Carlo Strapparava, *ITC-irst*

Humor is essential to communication. It relates directly to themes such as entertainment, fun, emotions, aesthetic pleasure, motivation, attention, and engagement, which many people in the field of intelligent user interfaces believe are fundamental for future computer-based systems. Furthermore, humans probably can't survive without humor. Computational humor has the potential to turn computers into extraordinarily creative and motivational tools. So, human-computer interaction must evolve beyond usability and productivity. Even though humor is complex to reproduce, it's realistic to model some types of humor production and to aim at implementing this capability in computational systems.

Humor, emotions, beliefs, and creativity

Humor is a powerful generator of emotions. As such, it affects people's psychological states, directs their attention, influences memorization and decision making, and creates desires and emotions. Actually, emotions are an extraordinary instrument for motivation and persuasion because those who are capable of transmitting and evoking them have the power to influence other people's opinions and behavior. Humor, therefore, allows for conscious and constructive use of the affective states it generates. Affective induction through verbal language is particularly interesting, and humor is one of the most effective ways of achieving it. Purposeful use of humorous techniques enables us to induce positive emotions and mood and to exploit their cognitive and behavioral effects.

Humor acts not only on emotions, but also on human beliefs. A joke plays on the hearer's beliefs and expectations. By infringing on them, it causes surprise and then hilarity. Jesting with beliefs and opinions, humor induces irony and helps people not to take themselves too seriously. Sometimes simple wit can sweep away a negative outlook that limits people's desires and abilities. Wit can help people overcome self-concern and pessimism that can prevent them from pursuing more ambitious goals and objectives.

Humor encourages creativity as well. The change in perspective that humorous situations create induces new ways of interpreting an event. By making fun of clichés and stressing their inconsistency, people become more open to new ideas and viewpoints. Creativity redraws the space of possibilities and delivers unexpected solutions to problems. Actually, creative stimuli constitute one of the most effective impulses for human activity. Machines equipped with humorous capabilities will be able to play an active role in inducing emotions and beliefs and in providing motivational support.

Background

Although researchers have been studying humor for a long time, including recent approaches in linguistics[1] and psychology, to date they have only made limited contributions toward the construction of computational-humor prototypes. Almost all the approaches try to deal with incongruity theory at various levels of refinement. Incongruity theory focuses on the element of surprise, stating that a conflict between what you expect and what actually occurs creates humor. This accounts for much of humor's most obvious features: ambiguity or double meaning.

Kim Binsted and Graeme Ritchie made one of the first attempts at automatic humor production.[2] They devised a formal model of the semantic and syntactic regularities underlying some types of puns. They then implemented the model in a system called JAPE (Joke Analysis and Production Engine) that could automatically generate amusing puns.

> Although researchers have been studying humor for a long time, to date they have only made limited contributions toward the construction of computational-humor prototypes.

The goal of our HAHAcronym project[3] was to develop a system that could automatically generate humorous versions of existing acronyms or produce new, amusing acronyms constrained to be valid vocabulary words, starting with user-provided concepts. The system achieved humorous effects mainly on the basis of incongruity. For example, it turned IJCAI—International Joint Conference on Artificial Intelligence—into Irrational Joint Conference on Antenuptial Intemperance.

Humor recognition has received less attention. Rada Mihalcea and Carlo Strapparava investigated the application of text-categorization techniques to humor recognition.[4] In particular, they showed that classification techniques are a viable approach for distinguishing between humorous and nonhumorous text, through experiments performed on very large data sets.

Applied scenarios: Humorous advertisements and headlines

Humor is an important way to communicate new ideas and change perspectives. On the cognitive side, humor has two important properties:

- It helps get and keep people's attention. The type and rhythm of humor can vary, and the time involved in building the effect can be different in different cases. Sometimes the context—such as joke telling—leads you to expect a humorous climax, which might only occur after a long while. Other times the effect occurs in almost no time, with one perceptive act—for instance, in static visual humor, funny posters, or in cases when some well-established convention is reversed with an utterance.
- It helps memory. For instance, we commonly connect knowledge we've acquired to a humorous remark or event in our memories. In learning a foreign language, an involuntary funny situation might occur because of so-called "false friends"—words that sound similar in two languages and might have the same origin but have very different meanings. The humorous experience is conducive to remembering the word's correct use.

No wonder humor has become a favorite strategy for creative people in the advertising business. In fact, among fields that use creative language, advertising is probably the area in which creativity (and hence humor) has the most precise objectives.

From an applied AI viewpoint, we believe that an environment for proposing solutions to advertising professionals can be a realistic practical application of computational humor and a useful first attempt at dealing with creative language.

Some examples of the huge array of opportunities that language offers and that existent natural-language-processing techniques can cope with include rhymes, wordplay, popular sayings or proverbs, quotations, alliteration, triplets, chiasmus, neologism, non sequitur, adaptation of existing expressions, adaptation of proverbs, personification, and synaesthesia (two or more senses combined).

The humorous variation of newspaper headlines is an important research direction, given the particular kinds of linguistic phenomena they display. Indeed, it's possible to exploit their elliptic nature, specific syntax, and deliberate use of rhetorical devices to create funny variations, leveraging lexical and syntactic ambiguity (for example, "Marijuana issues sent to joint committee"[5]).

Resources and processing

Our work aims at applicability in an unrestricted domain (the application is nonetheless limited in scope; no general mechanism encompasses all kinds of humor). Our approach relies on standard, non-humor-oriented resources (with some extensions and modifications) and the development of specific algorithms that implement and elaborate suitable linguistic-humor theories.

Resources

For example, in developing HAHAcronym, we used specialized thesauri, repositories, and in particular WordNet Domains, an extension developed at ITC-irst of the well-known English WordNet. WordNet Domains annotates synsets (that is, synonym sets) with subject field codes (or domain labels)—"Medicine," "Architecture," and so on. For HAHAcronym, we modeled an independent structure of domain opposition, such as "Religion" versus "Technology," "Sex" versus "Religion," and so on as a basic resource for the incongruity generator.

In our current work, we use a pronunciation dictionary, lexical knowledge bases including WordNet and WordNet Domains, WordNet-Affect (an affective lexical resource based on an extension of WordNet), a grammar for acronyms, a repository of commonsensical sentences, a database of proverbs and clichés, and idioms. Algorithms for making use of these resources include traditional components such as parsers, morphological analyzers, and specific reasoners.

Creating the affective similarity mechanism

An important component to include in such algorithms, especially for the kind of expressions we're dealing with, is the affective similarity mechanism. All words can potentially convey affective meaning; even apparently neutral ones can evoke pleasant or painful experiences. Although some words have emotional meanings with respect to individual stories, for many others, the affective power is part of the collective imagination (for example, "mom," "ghost," and "war").

So, measuring a generic term's affective power is important. To this end, we studied the use of words in textual productions, and in particular their co-occurrences with words having explicit affective meanings. We must distinguish between words directly referring to emotional states (such as "fear" and "cheerful") and those that indirectly refer to emotion depending on context (for example, words that indicate possible emotional causes such as "monster" or emotional responses such as "cry"). We call the former *direct affective words* and the latter *indirect affective* words.

We developed an affective similarity mechanism[6] that consists of

- the organization of direct affective words and synsets inside WordNet-Affect and
- a selection function (called *affective weight*) based on a semantic similarity mechanism. This similarity mechanism is automatically acquired in an unsupervised way from a large corpus of texts (hundreds of millions of words) to individuate the indirect affective lexicon.

All words can potentially convey affective meaning; even apparently neutral ones can evoke pleasant or painful experiences. Many words' affective power is part of the collective imagination.

For example, if we input the verb "shoot" with negative valence, the system individuates the emotional category of horror. Then it extracts the target-noun "gun" and the causative evaluative adjective "frightening" and finally generates the noun phrase "frightening gun."

Using the mechanism to produce humor

In most AI fields, researchers have understood the difficulties of reasoning on deep world knowledge for a while. A clear problem exists in scaling up from experiments to meaningful large-scale applications. This is even more obvious in areas such as humor, where good quality requires a subtle understanding of situations and is normally the privilege of talented individuals. In this sense, we can refer to computational humor as an *AI-complete* problem. Our goal is to produce general mechanisms for humorous revisitation of verbal expressions, to work in unrestricted domains.

Our approach can be synthesized in the following elements.

- We tend to start from existing material—for instance, well-known expressions or newspaper headlines—and produce some *optimal innovation* characterized by irony or another humorous connotation.
- The variation is based on reasoning on various resources, such as those we described earlier. No deep representation of meaning and pragmatics exists at the complex-expression level. However, there is more detailed representation at the individual-component level, permitting some type of reasoning (an example is the affective lexical resources indicated earlier where lexical reasoning is strongly involved). For more complex expressions, we adopt more opaque but dynamic and robust mechanisms based on learning and similarity, such as the one we mentioned earlier. The potential is particularly strong in dealing with new material, such as reacting to novel expressions in a dialogue or, as we are working on now, making fun of fresh headlines.
- The system architecture accommodates numerous resources and revision mechanisms. The coordination of the various modules is based on a general optimal innovation principle and numerous specific humor heuristic mechanisms inspired by the General Theory of Verbal Humor.[1]

We're working on postprocessing based on mechanisms for automatic humor evaluation; this is an evolution of humor recognition work we mentioned in the "Background" section. Right now, however, our system produces results in the form of a set of proposed expressions. The choice among them is left to the human user; few things are worse than poor humor, so it's better not to risk it.

References

1. S. Attardo, *Linguistic Theory of Humor*, Mouton de Gruyter, 1994.
2. K. Binsted and G. Ritchie, "Computational Rules for Punning Riddles," *Humor: Int'l J. Humor Research*, vol. 10, no. 1, 1997, pp. 25–76.
3. O. Stock and C. Strapparava, "Getting Serious about the Development of Computational Humor," *Proc. 18th Int'l Joint Conf. Artificial Intelligence* (IJCAI 03), Morgan Kaufmann, 2003, pp. 59–64.
4. R. Mihalcea and C. Strapparava, "Laughter Abounds in the Mouths of Computers: Investigations in Automatic Humor Recognition," *Proc. Intelligent Technologies for Interactive Entertainment* (INTETAIN 05), LNCS 3814, Springer, 2005, pp. 84–93.
5. C. Bucaria, "Lexical and Syntactic Ambiguity as a Source of Humor," *Humor: Int'l J. Humor Research*, vol. 17, no. 3, 2004, pp. 279–309.
6. A. Valitutti, C. Strapparava, and O. Stock, "Lexical Resources and Semantic Similarity for Affective Evaluative Expressions Generation," *Proc. 1st Int'l Conf. Affective Computing and Intelligent Interaction* (ACII 05), LNCS 3784, Springer, 2005, pp. 474–481.

The STANDUP Interactive Riddle-Builder

Graeme Ritchie, *University of Aberdeen*
Ruli Manurung and Helen Pain, *University of Edinburgh*
Annalu Waller and Dave O'Mara, *University of Dundee*

As children grow up, their language and communication skills develop as a result of their experience in the world and their interactions with other language users. In many cultures, an important type of interaction is language play with the child's peer group—word games and joke telling. A child with a disability such as cerebral palsy might converse using a voice output communication aid—a speech synthesizer attached to a physical input device. This cumbersome way of talking tends to isolate a child from the repartee, banter, and joke telling typical of the playground. This lack of practice can inhibit the development of language skills, leading to a lack of conversational fluency or even an undermining of social skills.

Our STANDUP (System to Augment Non-speakers' Dialogue Using Puns) project aims to take a small step toward alleviating this problem by providing, in software, a language playground for children with disabilities. The program provides an interactive user interface, specially designed for children with limited motor skills, through which a child can create simple jokes (riddles based on puns) by selecting words or topics. Here are two typical jokes that the system produces:

- What kind of berry is a stream? A current currant.
- How is an unmannered visitor different from a beneficial respite? One is a rude guest, the other is a good rest.

The system isn't just an online jokebook. It builds new jokes on the spot using 10 simple patterns for the essential shapes of punning riddles and a lexical database of about 130,000 words and phrases. We hope children will enjoy using the software to experiment with sounds and meanings to the benefit of their linguistic skills.

> The program provides an interactive user interface, specially designed for children with limited motor skills, through which a child can create simple jokes by selecting words or topics.

Background

Although various researchers have attempted since 1992 to get computers to produce novel jokes,[1] STANDUP's main predecessor is the JAPE system.[2] That program could churn out hundreds of punning riddles, some of which children judged to be of reasonable quality. However, it was only a rough research prototype; it took a long time to produce results and had no real user interface, and the ordinary user couldn't control it. We've used JAPE's central ideas to create a fully engineered, large-scale, interactive riddle generator with a user interface specially aimed at children with disabilities.

Designing with users

As computational humor is still in its infancy, we had no precedent for real-world use of a system such as STANDUP and certainly no experience of providing a joke generator for children with disabilities. We therefore devoted a substantial portion of the project to user-centered design. We consulted potential users and associated experts (teachers and speech and language therapists) about how the system, particularly the user interface, should operate.[3,4] In the early stages, we used nonsoftware mock-ups. We showed users laminated sheets representing screen configurations and asked them to step through tasks by pointing to the buttons in the pictures. The experimenter responded by replacing each sheet with the appropriate next screen shot. We adopted this low-tech approach to emphasize to the participants that the system hadn't yet been built and that suggestions or criticisms at this stage could genuinely influence the software's eventual design. Experience had shown that software mock-ups, particularly if very slick, give the impression that a working program is already available. This discourages participants from asking for changes and can even distract them into asking how they can get hold of this apparently working program.

On the basis of our studies' results, we built a software mock-up of the user interface (with a dummy joke generator) and tested it for usability with suitable users. Teachers and therapists again gave their advice.

In parallel with this, we designed and implemented the joke-generating back end. This incorporated a wide variety of facilities for manipulating words and joke structures, but our studies with users and experts suggested that this particular user group needed only a subset of these.

How the system works

You can view the STANDUP program as having two relatively separate major parts: the front end, which embodies the user interface and controls any user options, and the back end, which manages the lexical database and generates the jokes.

The user interface displays three main areas: the general navigation bar, the joke-selection menu, and the progress chart (see figure 3). The navigation bar is a standard set of buttons for going back, forward, exiting, and so on. The joke-selection menu consists of large labeled buttons through

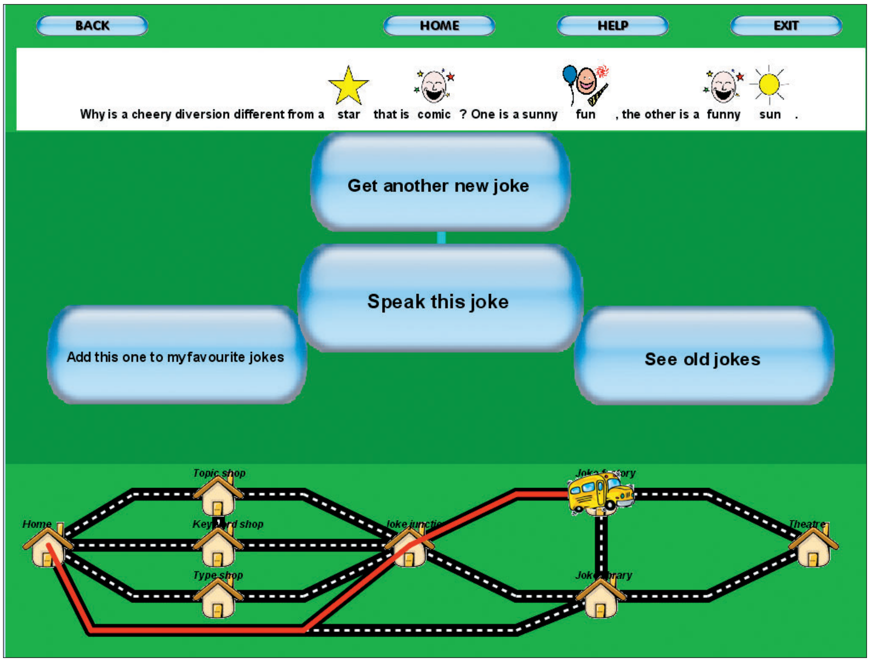


Figure 3. The STANDUP user interface.

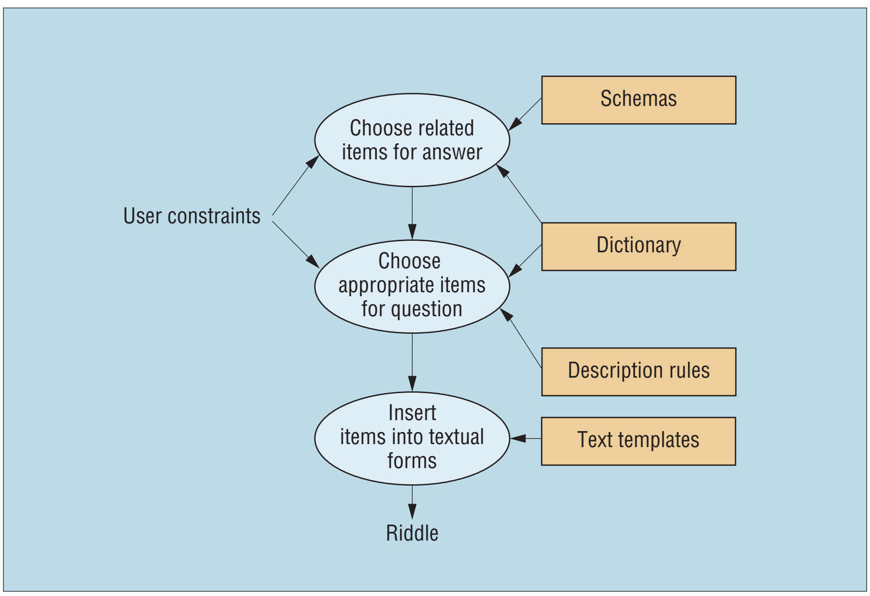


Figure 4. The structure of the STANDUP back end.

which the user controls the joke generator using a standard mouse, a touch screen, or a single-switch scanning interface (routinely adopted for those with limited motor skills). The progress chart shows where the user is in creating or finding a joke, using the metaphor of a bus journey along a simple road network, with stops such as "Word Shop" and "Joke Factory."

The back end (see figure 4) contains several components: a set of schemas, a set of description rules, a set of text templates, and a dictionary. The schemas define the linguistic patterns underlying punning riddles. For example, a schema might represent information such as

Find items X, Y in the dictionary such that X is a noun or adjective, Y is a noun, and X and Y sound the same.

This describes the central relationships in an example like the current/currant joke shown earlier.

The schema contains information that mainly constrains the ingredients of the riddle's answer, because that's where the pun occurs in all the types of riddle we've used. Once the system finds suitable items to match a schema, it passes these dictionary entries on to the description rules, which flesh out the descriptive phrases needed for the question. For example, starting from "rude" and "guest," it might try to find a way to build a phrase describing or meaning the same as "rude guest," such as "unmannered visitor," or it might select two items that can be "crossed" to produce "rude guest," such as "boor" and "visitor."

For the third stage, text templates contain canned strings such as "What do you get when you cross a ... and a …" or "What do you call a ..." alongside labeled slots into which the template-handling software slots the words and phrases provided by the schemas and the description rules, thus producing the final text.

The user can control this process via the graphical interface by imposing constraints on answer building (the schema), question building (the description rules), or both. For example, the user might specify that the joke must contain a particular word, be on a particular topic, or be a particular type of riddle.

Joke creation depends heavily on the dictionary, which contains information from numerous sources in a relational database. We took syntactic categories (such as noun and verb) and semantic relations (synonymy, being a subclass of, and so on) from the public-domain lexicon WordNet,[5] which contains approximately 200,000 entries. For information about the sound of words, we used the Unisyn pronunciation dictionary to turn a word or phrase's ordinary spelling into a standard phonetic notation. For our final user trials, we attached pictures to as many of the words as possible, using proprietary symbol sets that two companies involved in creating software aids for disabled users lent to us. The resulting lexical database of around 130,000 entries contains several tables, each representing one important relationship (such as between a word and its pronunciation or a word and its synonyms). In this way, we were able to implement dictionary searches as SQL database queries.

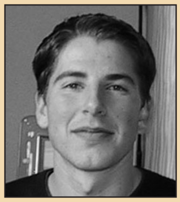
Although the joke-generation ideas are relatively simple and have already been tested in principle in the JAPE project, designing and implementing a large-scale, efficient, robust, easily usable system involved considerable work. We tried to make our designs as general as possible and to automate as much of the dictionary creation as possible, so that it should be relatively easy to create revised versions of the dictionary (for example, from a new version of WordNet). We also hope to make the lexical database available to other projects.

How will children use it?

We're about to start our final evaluations of the full system with users. This will involve visiting schools in the surrounding area, both special-needs establishments and mainstream schools. There we shall see how children—both with and without language-impairing disabilities—use the system. Beforehand, we'll assess certain aspects of each child's literacy to give us a context for interpreting what we observe. The time available to us (a few months) isn't sufficient for a long-term study of the software's effects. However, we'll carry out some tests at the end of a child's exploration of the system to see if the sessions have been beneficial in any way.

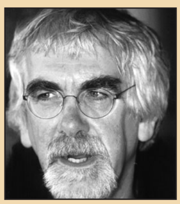
This project is very much an exploration of possibilities, and we don't know what we'll find out. However, we hope that the work will help move computational humor from tentative research to practical applications. In particular, a software language playground like this could well be of wider use in educational settings. ■

Kim Binsted is an assistant professor in Information and Computer Sciences at the University of Hawaii. Contact her at binsted@hawaii.edu.

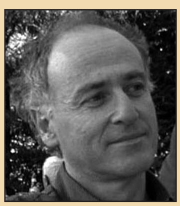


Benjamin Bergen is an assistant professor in the Linguistics Department at the University of Hawaii, Manoa, where he heads the Language and Cognition Laboratory. Contact him at bergen@hawaii.edu.

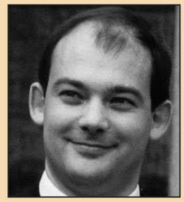
Seana Coulson is an associate professor in the Cognitive Science Department at the University of California, San Diego, where she heads the Brain and Cognition Laboratory. Contact her at coulson@cogsci.ucsd.edu.



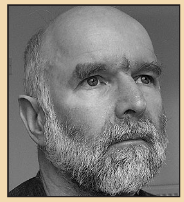
Anton Nijholt is a professor and chair of human-media interaction in the University of Twente's Department of Computer Science. Contact him at anijholt@cs.utwente.nl.



Oliviero Stock is a senior fellow at and former director of ITC-irst (Center for Scientific and Technological Research). Contact him at stock@itc.it.



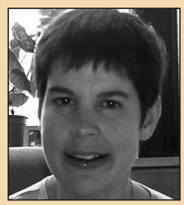
Carlo Strapparava is a senior researcher at ITC-irst in the Communication and Technologies Division. Contact him at strappa@itc.it.



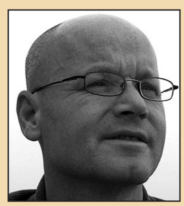
Graeme Ritchie is a senior research fellow in the University of Aberdeen's Department of Computing Science. Contact him at gritchie@csd.abdn.ac.uk.

Ruli Manurung is a research fellow in the University of Edinburgh's School of Informatics. Contact him at ruli.manurung@ed.ac.uk.

Helen Pain is a senior lecturer in the University of Edinburgh's School of Informatics. Contact her at h.pain@ed.ac.uk.



Annalu Waller is a lecturer in the University of Dundee's Division of Applied Computing. Contact her at awaller@computing.dundee.ac.uk.



Dave O'Mara is a postdoctoral research assistant in the University of Dundee's Division of Applied Computing. Contact him at domara@computing.dundee.ac.uk.

Acknowledgments

Grants GR/S15402/01 and GR/R83217/01 from the UK's Engineering and Physical Sciences Research Council supported this work. We're grateful to Widgit Software and Mayer-Johnson LLC for their help.

References

1. G. Ritchie, *The Linguistic Analysis of Jokes*, Routledge, 2004.
2. K. Binsted, H. Pain, and G. Ritchie, "Children's Evaluation of Computer-Generated Punning Riddles," *Pragmatics and Cognition*, vol. 5, no. 2, 1997, pp. 309–358.
3. R. Manurung et al., "Facilitating User Feedback in the Design of a Novel Joke Generation System for People with Severe Communication Impairment," *Proc. 11th Int'l Conf. Human-Computer Interaction* (HCI 05), CD-ROM, Lawrence Erlbaum, 2005.
4. D. O'Mara et al., "The Role of Assisted Communicators as Domain Experts in Early Software Design," *Proc. 11th Biennial Conf. Int'l Soc. for Augmentative and Alternative Communication*, CD-ROM, ISAAC, 2004.
5. C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.