Talk Notes

So all of this is just draft stuff to give you something to go on. Change whatever you like. Present the material as you see fit.

The talk is 20 minutes. I reckon there's roughly the right amount of material here, but feel free to cut bits if you like. NB: Don't spend too long on the introductory sections (this is the mistake I always make), try to blast through them at a brisk pace.

Slide 1

Hello, etc.

Intro:
Laughter is an intrinsically human thing. Comedy plays an important role in social relations, and a sense of humour is a key part of a person's character.

An A.I. theory of humour is therefore an attractive goal. Firstly, for the light it could shed on this interesting aspect of human behaviour. Also, the ability to make and recognise jokes could aid Human Computer Interaction in many areas.

Creativity is key to humour. One might suppose that a humorous machine could be created simply by storing examples, coupled with techniques for recognising jokes or selecting jokes relevant to the current context. This is unlikely to work well at either task. Originality is an important part of humour (hence Sheridan's classic insult "The right honourable gentleman is indebted to his memory for his jests, and to his imagination for his facts."), and jokes quickly go stale.

We have looked for logical structures that give rise to jokes, structures suitable for computer generation. We therefore focuses on structures that require relatively little world-knowledge.

In this talk, I'd like to present something of the framework we've developed, plus some examples of joke analysis. I'll skip over the formal aspects of the work – which the paper covers.

Computational Humour

There has been a lot of work on humour from a psychological perspective, but very little from an AI perspective.

Not using computers in humour. There are a number of systems that use computers to produce comedy. Most of these systems work by randomly mangling text (e.g. Putting text into an exaggerated dialect) or randomly combining text fragments (e.g. The Postmodernism Generator, which spits out wonderful nonsense essays). These systems create humorous nonsense. The results can be very funny. However these systems are not steps on the way to computational humour. They do not contain a model of humour, have no ability to recognise humour, and require considerable hand-coding of rules to create. Such systems embody the creativity of their author, rather than giving creativity to the computer.

So, by computational humour we mean systems that have a *model* for makes a joke, plus some *algorithms* for using that model in *generation* or *recognition*.

There is – as far as we know – really only one previous attempt at producing such a system: Kim

Binsted and Graeme Ritchie's JAPE back in 1994 (which has since been extended by other people). They looked at jokes based on ambiguity. Identified puns as a computationally tractable class of jokes. Produced a program – JAPE - which produces riddle-puns such as:
"What's the difference between a dog and a painter?
One can shed a coat and the other can coat a shed.".

Issue a Disclaimer:
This is **not** meant to be a complete theory of humour. We are only covering a narrow range of jokes.
We focus on abstract forms of joke and witty comments - which may be more amenable to computer generation. We identify several logical forms that give rise to jokes. By logical form, I mean an abstract form – something that can be expressed as a formula. These then suggest algorithms for joke generation – though these have yet to be implemented and tested.

Intentionally Poor Communication

Now in 'normal' speech – by which I mean clear and simple communication – we obey certain rules. This clear speech is sometimes called *Gricean* communication, after the linguist Grice who did some very influential work on the rules of 'normal' speech.

The basic idea we explored is that there is a class of joke related to poor speech.
Initially this seems strange: Wit is perhaps the very opposite of poor speech – yet it often involves breaking Grice's rules.
Because jokes are playful. They break the rules.

Let me give you a couple of examples of such jokes:

Jewish Mother: Help, help! My son, the doctor is drowning!
This breaks Grice's rule that you should not make be more informative than necessary

Every minute, somewhere in the world a woman gives birth. We must find this woman & stop her
This breaks Gricean rule that you should avoid ambiguity – in this case parsing ambiguity.

Now, poor speech alone is rarely funny. Wit should be deliberately poor.
There are parallels here with deliberate comic stupidity, and the deliberate clumsiness of clowning. If possible, wit should be surprisingly ingeniously brilliantly poor. Typically this is achieved by initially looking normal, whilst signalling an intention to be ridiculous . So wit consists of cleverly disguised obviously poor speech.

This intention forms an interesting but vague constraint. It can be conveyed by: by tone of voice (e.g. sarcasm), by context, by the character of the speaker (or in the case of mockery, the character of the target), by contrast with similar well-known utterances, by cross-reference, by self-reference, or by the artfulness of the joke's construction. The unifying factor is that all of these signs distinguish the joke from a genuinely poor utterance.

Forms of Joke

We give a logical definition for poor speech, and then we look at forms of joke that break this. That allows us to analyse a range of jokes, and from that we identify a number of *joke forms* which we can express as formulae. We can categorise these forms:

● Playful Ambiguity: e.g. Puns
● Obvious Tautologies: what you say is so obviously true, that it amounts to saying nothing
E.g. (1 - stating part of the definition as if it's an interesting fact): *"Almost all acts of self-abuse are committed by someone who knows the victim, and these cases are hardly ever reported."*
(2 - more blatant 'x or not x') *"Sometimes it feels like my soul is being sucked out through my arse. And sometimes, it doesn't."*

● Contradictions:
eg1: *There are 3 types of mathematician: those who can count and those who can't*
eg2: Mathematician turned comedian Tom Lehrer: *The Harvard mathematics department was a hotbed of celibacy.*

● Convoluted statements: see a couple in a minute

● And more...

● Quantifier Abuse: Using universal quantifiers when there aren't many (or even any) examples
E.g. Henry Ford's *"You can have any colour you like as long as it's black."*
e.g. 2 *"They had nothing. But they were willing to risk it all."* (tagline from a film The Committments)

These type of jokes give us formulae such as this one (which reads, forall a in set X, P(a) is true, but set X is of size 0 or 1).
To use such a formula in generation, we need ways of instantiating the variables, and evaluating relations. Then – because this is at the semantic level – we need either a template or a simple grammar for presenting the joke (i.e. converting it into a proper sentence). We'll look a bit more at generation in the next slide

Some (very) Early Results

This work is very much at a preliminary stage. However we have implemented generators for a couple of the joke forms.

Sarcasm (this is a form of convoluted statement, saying things in a less clear manner than you could)

Our sarcasm system converts from plain-talking text into sarcastic text, or vice versa. Here is an example.

Italics indicate sarcasm

This system uses

a shallow parser for identifying adjectives,

a thesaurus for finding antonyms.

We use the MRC to choose amongst the antonyms. The MRC is a database that measures certain properties of words, such as their 'image strength', which is done according to a statistical analysis of a large corpus of text.

This system is simple, but seems fairly robust.

Inappropriate metaphors (also a form of convolution)

This is where you say "As P as a Q" - where *Q* is an example of not *P*ness – as a convoluted way of

saying “not P”

Here we use a thesaurus to identify antonyms,

then we use LSA to find examples

LSA compares word-usage; it's a corpus-based statistical comparison, to identify examples of the antonym

We filter potential examples with WordNet to give nouns of the correct type. WordNet is a kind of enhanced thesaurus.

This system is less robust. These are some good examples – we could also give some bad examples here.

Future Work

This is preliminary work, and there is quite a bit to be done. Anyone who is interested in being involved in this project, please do contact us. This is a sideline for us, and we don't have the time we'd like to devote to it.

- Implementation & Testing – of the proposed algorithms
- More joke forms - The ones we identify is not a complete set.
- Richer theory of discourse – We've assumed for simplicity that there are only three types of speech acts (information exchange, mistakes, and jokes). A richer theory of discourse - one containing more speech acts, and relating these to human interaction - would allow us to filter out some non-jokes, and should allow us to identify more joke forms.
- Formalisation

Our paper gives a semi-formal account. A proper *formal* formalisation is desirable, both for the concreteness it would give the theory, and because it would allow:

- Automatic discovery of joke forms

The joke forms we've identified can all be derived from one axiom, defining deliberately poor communication. We could look at *automatically* driving these joke forms - and others - from such axioms.

Thank You

(with useful resources and selected references)
Conclusion:

We have found a relatively abstract form of humour that could be computationally tractable. This is the class of jokes and witty statements which draw their humour from deliberately breaking the rules of normal conversation – a class of humour we summarise as ingeniously poor communication.

The work presented here is in a preliminary stage, and there is clearly much more to be done. It contains many limitations – this is a very difficult field for AI, being so inimicably human. That's the thing: the things that make it hard also make it an important and interesting field, one which has perhaps been overly neglected.

Hopefully the algorithm schemes we have described here will prove successful when implemented. We'll keep you posted.

Thank you.