

What is Poorly Said is a Little Funny

Jonas Sjöbergh and Kenji Araki

Graduate School of Information Science and Technology
Hokkaido University
Sapporo, Japan
{js, araki}@media.eng.hokudai.ac.jp

Abstract

We implement several different methods for generating jokes in English. The common theme is to intentionally produce poor utterances by breaking Grice's maxims of conversation. The generated jokes are evaluated and compared to human made jokes. They are in general quite weak jokes, though there are a few high scoring jokes and many jokes that score higher than the most boring human joke.

1. Introduction

Humor is common in interactions between humans. Research on humor has been done in many different fields, such as psychology, philosophy, sociology and linguistics. For a good introduction and overview of humor in automatic language processing, see (Binsted et al., 2006). When it comes to computer implementations, two main approaches have been explored. One is humor generation, where systems for generating usually quite simple forms of jokes, e.g. word play jokes, have been constructed (Binsted, 1996; Binsted and Takizawa, 1998; Yokogawa, 2001; Stark et al., 2005). The second is humor recognition, where systems to recognize whether a text is a joke or not have been constructed (Taylor and Mazlack, 2005; Mihalcea and Strapparava, 2005).

This paper belongs in the first group. We try to generate a few different types of jokes. The main inspiration has been a paper with ideas on possible ways to generate jokes by breaking Grice's maxims of communication (Winterstein and Mhatre, 2005). To the best of our knowledge, most of these ideas have not been implemented before. We implement and evaluate some of them and some other similar joke generation methods.

2. Inspirational Theory

In (Winterstein and Mhatre, 2005) several ideas for generating humorous utterances are presented (though only one was actually implemented and no evaluation was done). They all constitute intentionally poor speech acts. Many types of jokes seem to fit into the intentionally poor speech category. The authors hypothesize that this could be a way of showing off mental dexterity, since creating these jokes often require intelligence and creativity. The hard part is not to produce poor utterances, which is easy, but to produce utterances that are clearly poor and at the same time signalling that this is done intentionally.

Several possible ways of generating jokes are presented, based on breaking Grice's maxims of communication. We here use the presentation of the maxims from (Winterstein and Mhatre, 2005), though the source is originally (Grice, 1975). An example would be the maxim of quantity, one part of which states that you should be as informative as necessary to make your point clear. Breaking this while showing that you are doing it on purpose can lead to jokes

like: "Recursive, adj.; see Recursive".

In this paper, we take the suggested joke generation strategies in (Winterstein and Mhatre, 2005) as a starting point and implement parts of them or related methods. We then evaluate the generated jokes.

3. Different Generation Strategies

Here we begin the presentation of the different methods we have implemented. They are grouped under the maxim of speech they are related to. All methods are fairly simplistic, generally using only readily available data sets and very simple algorithms. They are intended more as exploratory attempts or proofs of concept than programs to generate state of the art humor (which seems very difficult with today's language technology).

The four subsections here relate directly to Grice's maxims, and the generation methods there are based on or inspired by the example jokes and discussions in (Winterstein and Mhatre, 2005). The next section deals with the more detailed suggestions from that paper, which are based on the formal analysis of poor utterances therein.

We have tried to make use of only readily available resources, to see if interesting jokes can be generated without large amounts of manual work. Of course, many of these resources are only available in English or very few other languages, so generating jokes in other languages may be more difficult. Automatic humor generation methods that report fairly successful results (for instance (Binsted, 1996)), as opposed to the "not that funny" results that seem to be the norm, often make use of quite detailed semantic lexicons and similar semantic resources. We have not used any resources with structured rich semantic information, since we did not have any available.

3.1. Maxim of Quantity

The two main rules of quantity are: "Make your contribution as informative as necessary" and "Do not make it more informative than necessary".

We generate jokes of a type we call "unnecessary clarification" in this category. An example would be "He stood there licking his own lips". The explanation that the lips he licked were his own is unnecessary, so the statement can be seen as humorous.

The jokes are generated by taking a list of idioms involving body parts from a collection of English idioms on the In-

ternet¹. These idioms come with example sentences, which we use, but examples could also be automatically gathered for instance from the Internet by searching for the idioms themselves. To generate a joke, an example of use of an idiom related to body parts is taken and checked for the occurrence of a pair like “I – my” or “we – our”. If such a pair occurs and the word “own” is not already present, it is added. Jokes can be generated by replacing “his lips” with “my lips” or “someone else’s lips” too, though this is no longer breaking the principle of avoiding unnecessary explanations. A more sophisticated method that recognizes when adding the word “own” is in fact a normal clarification (thus does not lead to a joke) would likely improve the results, since this seems to be quite common. Example generated joke: “You could see she was hurt - she wears her *own* heart on her sleeve.”

3.2. Maxim of Quality

The maxim of quality states: “Do not say things that you believe to be false, or for which you lack adequate evidence”.

In this category we generate sarcastic answers to questions about places. We call this type the “false explanation”. This follows the pattern presented in (Winterstein and Mhatre, 2005) that quotes the movie *Casablanca*: “What brought you to *Casablanca*?” “My health. I came to *Casablanca* for the waters.” “The waters? What waters? We’re in the desert.” “I was misinformed.”

These are generated by taking a list of place names, the 47 largest cities according to population size, from Wikipedia².

For a place name, Internet searches are then performed using the query “why are there no * in <place name>”. The most common text string matching the query is then considered to be a typical fact about this place. Then a joke can be created by filling out a pattern like the one from *Casablanca*. In general, this method seems too simplistic to generate interesting jokes. Example of a generated joke: “Why did you come to Hong Kong? Because of the spectacular *Swedes* here. But there are no *Swedes* here in Hong Kong?! You don’t say?”

Also as suggested in (Winterstein and Mhatre, 2005), we generate the “snotty answer” type jokes. These are of the form “Are you a computer?” or a similar question, followed by “No, I am a <adjective> <noun>”. An adjective and a noun is randomly drawn from the words in WordNet (Miller, 1990; Miller, 1995), thus producing a nonsense answer which can be funny. It is a one trick pony though, so while generating many of these jokes is easy it is not funny. It can be adapted to many other types of questions though, so another idea might be to write a program that detects when what kind of snotty answer can be used. Example joke: “Were these jokes made by a computer? No, by an undersexed dimash.”

3.3. Maxim of Relevance

The maxim of relevance simply states: “Be relevant”. The “snotty answer” method in the previous section also vio-

lates the “be relevant” principle.

We have also implemented the “strange act – apology” method. It first selects a word signifying a place by looking through the Open Mind Common Sense (Singh, 2002) data for statements like “<place> is a place”. It then collects things related to such a place by looking for statements of the form “[you can] find a <thing> in a <place>”. If at least two things are found, an action appropriate for the place is found using an Internet search for “you can * in a <place>”. A strange action is then found similarly, using “you can’t * a <thing₁>”. These are then all put together in a pattern like “A man came to a <place> to <action>. He then started to <strange action> a <thing₁>. Then he said ’I am sorry, I thought it was a <thing₂>’”. An example of a generated joke: “A man came to a mall to do anything. He then started to close a shop. Then he said ’I am sorry, I thought it was a barbershop’.”

Other than this, we have not implemented any methods for this maxim. It is generally easy to produce things that are not relevant, but remaining understandable enough to be funny is harder.

3.4. Maxim of Manner

The maxim of manner states: “Avoid obscurity of expression”, “Avoid ambiguity”, “Be brief”, and “Be orderly”.

Breaking the maxim of manner seems the easiest, so there are several methods in this group. First we have the “strange metaphor”. It generates new metaphors by taking an adjective from WordNet. Then antonyms of this adjective are collected from dictionary.com³. Taking the original adjective, the Internet is searched for expressions of the form “as <adjective> as a *”. Then metaphors are generated by taking random antonyms and random matches to the search query, and outputting “as <antonym> as a <answer>”. This makes for an unusual expression, thus violating the “avoid obscurity” rule. Example: “stiff as a rag”.

Euphemisms are also a good source of humor. It is very common to find veiled references to taboo subjects in jokes. This can be seen as a play on the ambiguity restriction. We have two euphemism related strategies “Wiktionary euphemisms” and “idioms and euphemisms”.

The first begins by taking the listing of euphemisms from the Wiktionary⁴. For all euphemisms that are longer than one word and contain a verb (checked by using the list of words in WordNet) or a verb in “-ing” form (i.e. actually a noun, but created from a verb), a joke is created. If there is a noun in the euphemism, the joke is made using a pattern like “I like <noun>s, shall we <euphemism>?”. So for the euphemism “see a man about a horse”: “I like horses, shall we see a man about a horse?” is created. If no noun is found but the verb is in “-ing” form, this in itself is used instead, so “pushing up the daisies” becomes “I like pushing, how about pushing up the daisies with me?” since “daisies” was not recognized as a noun (though that could easily be fixed and would have made a better joke).

The second euphemism method uses a list of about 3,000 idioms and proverbs collected by searching the Internet for

¹<http://www.idiomconnection.com/>

²http://en.wikipedia.org/wiki/Main_Page

³<http://dictionary.reference.com/>

⁴<http://en.wiktionary.org/wiki/Category:Euphemisms>

“English idioms” and taking the first few pages that actually contained collections of idioms or proverbs. It also uses a list of about 2,500 “dirty word” expressions in English, collected by George Carlin⁵. Any dirty word expression of precisely two content words can be used. Idioms are checked for the occurrence of one of these two words, and jokes are created by replacing this word with the two word expression. Example of a generated joke: “a good man in an evil society seems the greatest villain of all” plus “skirt man” gives “a good ‘skirt man’ in an evil society seems the greatest villain of all”.

The “convoluted statement” method breaks the “be brief” principle, also using the idioms and proverbs. For idioms of at least four words, jokes are generated by replacing the original words with their definitions, normally composed of many words. Definitions are taken from the Internet, using the search query “definition of <word>”. Example result: “can’t organize or be responsible for a stick of wax with a wick in the middle (bureaucratic speech for ‘can’t hold a candle’)”.

For the “be orderly” principle, we have the “mucking fuddled” method. It searches the Internet for “I hate this <bad word> *” and then changes the first letter of the bad word and a following word. If at least one of the two new words is an existing word, it is considered an acceptable reordering of letters. An example is: “I hate this ducking fouchebag”. Another method that breaks the “be orderly” principle is “combination of innocent words”. It takes multi-word expressions from the list of dirty words and checks if they consist solely of words that do not occur by themselves in the dirty word list. If they also contain at least two content words, a joke of the form “None of the words <list of content words> are bad words, yet talking about <dirty expression> is taboo”. The first and last words from the multi-word expression are also interchanged when listing the words, so as not to give away the punch line too quickly. An example result is: “None of the words anaconda and trouser are bad words, yet talking about a trouser anaconda is taboo.”

A common problem with all these methods is that the references are often too obscure. For instance if an idiom or euphemism is not known or understood by the reader, the humor is lost.

4. Maxim of Normal Communication

In (Winterstein and Mhatre, 2005) the authors formalize what constitutes a poor utterance. They then present the maxim of normal communication, which is a combination of elements from the maxims of quantity and manner. It states that “In normal communication, poor utterances only occur by mistake”. An utterance is poor if given the knowledge one has and the utterance, it is easy to deduce another utterance with the same meaning but which is much less complex (e.g. a clearer way to express the same thing).

The authors then logically analyze what types of statements lead to poor speech acts. For these, they suggest some possible ways to generate such statements to make jokes. We here list the main approaches and present our implementations based on these.

4.1. Obvious Tautologies

Tautologies are poor utterances, and several ways to generate them are suggested. We have implemented two, “stating definitions” and “to be or not to be”. The “stating definitions” method simply takes a noun at random from WordNet, and produces “I like <noun>, especially <definition of noun>”. The definition is produced as for the “convoluted statement” method, i.e. using a web search for “definition of <noun>”. Example output: “I like infringements, especially if it is an act that disregards an agreement or a right”.

The “to be or not to be” method produces statements of the form “I sometimes feel like <adjective> <noun>, and sometimes I don’t” with small variations. Words are drawn at random from WordNet. Example: “Sometimes I feel like a well-lighted and light-green long pillow, but often times I don’t.”

As mentioned already in the original paper (Winterstein and Mhatre, 2005), tautologies are too easy to generate to be very funny in themselves. Other factors seem to be necessary to make these types of jokes funny, for instance word play or veiled references to taboo subjects.

4.2. Quantifier Abuse

When using quantifying expressions (or logical quantifiers in the more formal analysis of the inspirational paper), these imply certain things. One example is the “for all” quantifier, which in normal conversation would imply the existence of multiple objects. A joke breaking this expectation (thus being an example of a poor utterance) is: “You can have any color you like, as long as it is black”.

We have implemented the quantifier abuse method “any color you like” for this type of abuse. It searches the Open Mind Common Sense data for statements on the form “<phrase₁> is a <phrase₂>”. It then simply prints statements like “You can have any <phrase₂> that you like, as long as it is <phrase₁>”. These are rarely funny, but could perhaps be used in certain situations and would then be easy to generate. Example sentence: “You can choose any part of Bill you want, as long as it is Bill’s nose”.

We have also implemented the quantifier abuse method “giant exception”. “Every <something> except <example>” implies that there are many things not covered by “<example>”. Breaking this expectation leads to jokes like “I can resist anything but temptation”. We generate jokes of this type by using the pattern “I like [<noun> or <to verb>], unless you mean <definition>”. Since this is too simplistic to be much fun in general, we take only such nouns or verbs that have more than one word sense in WordNet and that also occur in the list of dirty words. Definitions are again fetched by searching for “definition of <expression>”. An example joke is: “I like a nice wand, unless we are talking about a rod used by a magician or water diviner”.

Just as for the “any color you like” method, the other types of quantifier abuse suggested seem hard to implement in a way that is funny.

⁵<http://www.georgecarlin.com/dirty/2443.html>

	Best Joke	Average	Human Level
Non-jokes	1.6	1.1	0
Real jokes	4.4	3.4	15
Unnecessary Clarification	3.0	1.9	2
False Explanation	2.7	2.2	5
Snotty Answer	2.5	1.9	3
Strange Act – Apology	2.4	1.8	1
Strange Metaphor	2.5	2.1	3
Wiktionary Euphemisms	2.7	2.1	4
Idioms and Euphemisms	2.2	1.6	1
Convolutd Statement	2.2	1.6	1
Mucking Fuddled	3.3	2.2	2
Combination of Innocent Words	2.8	2.4	5
Stating Definition	1.8	1.2	0
To Be or Not To Be	3.0	2.1	2
Giant Exception	2.6	1.8	1
Any Color You Like	2.3	2.0	4
Oxymoron	2.4	1.8	3

Table 1: Evaluation scores of the different types of jokes.

4.3. Contradictions

Contradictions are poor speech but generating really funny contradictions seems to be very difficult. The discussed joke type of joining conflicting statements is also as the authors note likely difficult to generate in such a way that it looks like a joke and not garbage.

The suggested “oxymoron” method is quite simple though. We generate oxymorons by taking “<adjective> <noun>” expressions or “<noun> of <noun>” expressions from WordNet. Then the oxymoron is generated by substituting one word for its antonym. The antonym is generated as in the “strange metaphor” method, by using dictionary.com. The results are often too unrecognizable to be funny by this method, so adjusting it to use mainly well known words and expressions would likely be a good idea.

4.4. Convolutd Statement

Generating unnecessarily detailed expressions from normal text is fairly straightforward, as described in the Convolutd Statement section of (Winterstein and Mhatre, 2005). Our implementation of this method was presented in the section on the maxim of manner.

The only method that is presented as implemented in (Winterstein and Mhatre, 2005) is a method for double negatives. It produces sarcasm of the form “input: You look terrible, output: [sarcasm] you look wonderful”. Since it was already implemented by the original authors we have not created an implementation of our own. It should be fairly straightforward to adapt the “oxymoron” or “strange metaphor” methods to produce sarcasm instead, should one wish to do so.

The final example mentioned in their paper is a method for generating inappropriate metaphors. Our implementation, “strange metaphor”, along those lines was presented in the section on the maxim of manner.

5. Evaluation

We evaluated the generation methods by generating five jokes with each of the fifteen methods. These were then presented together with fifteen real jokes from a corpus of oneliner jokes written by humans. Also for comparison purposes, five sentences from a corpus of normal text (i.e. not jokes) were also included.

These sentences were then presented in random order and readers were asked to grade how funny they thought the jokes were on a scale from 1 (not funny at all) to 5 (very funny). It was also possible to skip some jokes if the evaluator so desired. The results of the evaluation are presented in Table 1.

The average score of the best joke of each category, as well as the average score of all the jokes in the category is shown. The number of jokes that scored higher than the lowest scoring human made joke is also reported. A total of six persons has participated in the evaluations so far, though the evaluation is still available online. Six evaluators is of course very low, so for our future experiments we plan to generate jokes in other languages where we have an easier time to find native speaker evaluators.

Though there are a few funny jokes, most of the joke methods produce mainly quite weak jokes. Quite a few jokes, 37 out of 75, do score higher than the most boring human made joke (score 2.0), though sadly no generated joke scored higher than the human average (3.4). Almost all types of jokes (the exception being “stating definition”) scored higher than the non-joke reference sentences. That the generated jokes are not that great is not surprising. Other joke generation systems that report success figures generally also achieve quite poor performance. A nice application of using faulty anaphora resolution to generate jokes for example achieved 15% recall and precision (Tinholt and Nijholt, 2007).

It may be possible to use the joke generation as a help-

ful tool for making good jokes though, since many of the jokes could be made quite a bit funnier by a little human intervention. Many jokes fail because of very obscure idioms/euphemisms/words (joke not understood), broken grammar (making things boring), or simplistic pattern matching giving unwanted matches from unexpected sentences. Many of these could easily be corrected by a human.

For creating jokes completely automatically, more work needs to be done. A method to guess how funny something is likely to be would be very helpful, since the quality of the jokes generated by several methods varies wildly. It can also be noted that the methods that produced the funniest jokes had fairly low average scores, while the methods with higher averages did not produce as funny top scoring jokes.

One thing that would be helpful is a measurement for how recognizable an idiom or expression is. Many of the jokes generated by taking euphemisms for sexual acts or similar things are too obscure to be recognized and thus not funny. Checking the number of occurrences in a large corpus such as the Internet might help in this regard, and we plan to try this in the future.

As mentioned, there are also problems with the generated jokes containing broken grammar, which makes them hard to read. Possibly this can be mitigated by using automatic grammar checking tools, and also by making the language generation procedure more sophisticated. Of course, by making the generation part more sophisticated, there is also a risk of getting jokes that are funny mainly because of the creativity of the program creator and not because of the actual things created by the program.

Most of the successful jokes contained references to taboo subjects but some jokes with no taboo connections also scored highly. While the non-taboo jokes could be funny, they seem to work only very few times. The first joke from a certain type is funny, but the rest are just more of the same. With dirty words involved, the funniness seems to stay a little longer, though there were comments from some of the evaluators that there is too little variation between jokes of the same type, so they get old very fast.

6. Conclusions

We presented several methods for generating jokes, all based on intentionally breaking Grice's maxims of conversation. All methods are easy to implement and require only commonly available resources (data and tools).

Most of the methods produce mainly very weak jokes, though about half the jokes were funnier than the most boring human made joke. Problems include broken grammar and references to obscure or hard to recognize words and expressions.

While small improvements to the joke generation seem easy to make, it seems harder to generate jokes consistently on the level of fairly good human made jokes. Using the current joke generation methods as a support tool for a user creating jokes seems feasible though.

Acknowledgements

This work was done as a part of a project funded by the Japanese Society for the Promotion of Science (JSPS).

7. References

- Kim Binsted and Osamu Takizawa. 1998. BOKE: A Japanese punning riddle generator. *Journal of the Japanese Society for Artificial Intelligence*, 13(6):920–927.
- Kim Binsted, Benjamin Bergen, Seana Coulson, Anton Nijholt, Oliviero Stock, Carlo Strapparava, Graeme Ritchie, Ruli Manurung, Helen Pain, Annalu Waller, and Dave O'Mara. 2006. Computational humor. *IEEE Intelligent Systems*, 21(2):59–69.
- Kim Binsted. 1996. *Machine Humour: An Implemented Model of Puns*. Ph.D. thesis, University of Edinburgh, Edinburgh, United Kingdom.
- Paul Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3:41–58.
- Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of HLT/EMNLP*, Vancouver, Canada.
- George Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.
- George Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.
- Push Singh. 2002. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, Palo Alto, California.
- Jeff Stark, Kim Binsted, and Benjamin Bergen. 2005. Disjunctive selection for one-line jokes. In *Proceedings of INTETAIN 2005*, pages 174–182, Madonna di Campiglio, Italy.
- Julia Taylor and Lawrence Mazlack. 2005. Toward computational recognition of humorous intent. In *Proceedings of Cognitive Science Conference 2005 (CogSci 2005)*, pages 2166–2171, Stresa, Italy.
- Hans Wim Tinholt and Anton Nijholt. 2007. Computational humour: Utilizing cross-reference ambiguity for conversational jokes. In *CLIP 2007*, Camogli, Italy.
- Daniel Winterstein and Sebastian Mhatre. 2005. Logical forms in wit. In *Computational Creativity Workshop of IJCAI 2005*, University of Edinburgh, Scotland, UK.
- Toshihiko Yokogawa. 2001. Generation of Japanese puns based on similarity of articulation. In *Proceedings of IFSA/NAFIPS 2001*, Vancouver, Canada.