Multilingual pun generation

Kim Binsted
Sony CSL
Tokyo, Japan

Graeme Ritchie
University of Edinburgh
Edinburgh, Scotland

Osamu Takizawa
CRL
Tokyo, Japan

Helen Pain
University of Edinburgh
Edinburgh, Scotland

Abstract

We have developed a formal model of puns, based on an analysis of a large number of punning riddles, such as those published in children's joke books. We have implemented that model in programs which generate puns of this genre in English and in Japanese, using linguistic information from general-purpose lexicons. For the English-language version of the system, we conducted an extensive evaluation using children as 'joke judges'. For the Japanese-language version of the system, we conducted a smaller evaluation using adults. Both evaluations indicate that the computer-generated puns are of comparable quality to those generated by humans. We give a detailed description of the model, a brief description of the English-language implementation and evaluation, and a fuller description of the Japanese-language implementation and its evaluation. We also discuss the implications of its success.

1 Introduction

Here we present a formal model of simple punning riddles. This common genre is typified by the jokes found in children's joke books; for example:

> Where do cats go when they die? *Purrgatory.* [Webb, 1978]

Our model identifies and describes the essential features of punning riddles. It has been implemented in two separate programs: one generates punning riddles in English; the other generates punning riddles in Japanese. Both systems have been formally evaluated, and the results indicate that their output texts are of comparable quality to jokes of the genre generated by humans (and published in joke books).

In this paper, we discuss the model and both implementations in some detail. We also describe the evaluation of the Japanese system, and discuss the significance of the results. The evaluation of the English system has been presented in detail elsewhere [Binsted *et al.*, 1997] so is described only briefly.

2 Ambiguity in puns

Many types of joke use linguistic ambiguity to mislead the audience (the listener or reader) to humorous effect. *Low–level* ambiguities are those which allow several different senses for a word or words in a spoken or written text. In texts that have *high–level* ambiguity, the senses of the individual words are not in question, only the interpretation of the whole text. Here we concern ourselves mainly with jokes based on low–level ambiguities; for convenience we shall refer to these informally as *puns*.

We have identified three kinds of low–level ambiguity in English:

spelling ambiguity: This is where one phonological form corresponds to two (or more) written forms and senses. For example, the phoneme sequence [s,ia,r,ia,l] could be written either as "cereal" or as "serial".[1]

pronunciation ambiguity: This is when one written form corresponds to two (or more) phonological forms and senses. For example, the written form "wind" corresponds to the phonological forms [w,ih,n,d] (n. moving air) and [w,ay,n,d] (v. twist).

word sense ambiguity: This is when one phonological form *and* one written form correspond to two or more senses. Whether the word is spoken *or* read, the listener/reader cannot tell which sense of the word is meant. For example, "bank" is word sense ambiguous, in that it has two senses: the side of a river, and a financial institution.

In English, both spelling and word sense ambiguity are fairly common, while pronunciation ambiguity is relatively rare. In Japanese, pronunciation ambiguity is almost non-existent, due to the regularity of pronunciation in the language. Word sense ambiguity is also quite rare, since chinese characters (kanji) are often part of the written form of a word, and these usually carry enough semantic information to disambiguate the text. Spelling ambiguity, however, is very common in Japanese. For example, the word pronounced /hasi/ [2] has (at least)

[1] We represent English phonemes using the ARPAbet system [Robinson, 1996]

[2] Here we use the Kunrei phonetic notation.

three different written forms, each with a different sense ('bridge', 'edge' and 'chopsticks').

It is useful in discussing riddles to introduce the idea of 'confusability'. If a text is ambiguous in one form, then its variations in other forms are *confusable*. For example, the three written forms of /hasi/ are completely confusable. Confusable segments are often substituted, juxtaposed or compared in jokes.

It is also useful to introduce the concept of 'similarity'. Two texts can be *similar* in written form, spoken form, or sense, although here we are most interested in similarity of spoken form. Texts that are sufficiently similar can be considered confusable — although determining how similar the texts have to be to be confusable is problematic.

3 Punning mechanisms in English

There are the three strategies used in English puns to exploit spelling or word sense ambiguity: *juxtaposition*, *substitution*, and *comparison*.

Juxtaposition is the most simple mechanism, simply placing confusable segments near each other and treating them as a normal construction. For example:

> What do you get if you cross a dog and a kangaroo? *A pooch with a pouch.* [Ertner, 1993]

Jokes which use juxtaposition alone are really very weak, as they do little more than demonstrate an unexpected phonological similarity.

Substitution works by substituting one confusable segment for another, as part of a larger text, and using the resulting text as if it were a sensible construction. For example, the word "purr" is confusable with the first syllable of "purgatory":

> Where do cats go when they die? *Purrgatory.* [Ertner, 1993]

Comparison explicitly compares two confusable texts, usually by asking for similarities or differences in the question part of the riddle. Positive comparison riddles ask for similarities, and negative comparison riddles ask for differences.

An example of a negative comparison riddle is:

> What's the difference between a short witch and a deer running from the hunters? *One's a stunted hag and the other's a hunted stag.* [Webb, 1978]

and a positive comparison riddle:

> How is a red–haired loonie like a biscuit? *They're both ginger nuts.* [Webb, 1978]

The negative comparison mechanism can be seen as a specialization of the juxtaposition mechanism. Both mechanisms compare similar-sounding text segments; in juxtaposition puns, however, the comparison is implicit. Similarly, the positive comparison mechanism can be seen as a specialization of the substitution mechanism, using word-sense ambiguity only. In English, however, the comparison riddle is a very common genre with distinct features (use of metathesis, stereotyped surface forms, etc), so we treat them separately here.

4 Overview of the model

All of the mechanisms described above construct non-lexicalized (i.e. not already in the lexicon) segments of text by manipulating (usually substituting or juxtaposing) lexicalized words and phrases. The resulting constructed text segment is then used in a question-answer pair as if it were a semantically sensible construction. The effective meaning of this construction is a based on the meanings of the lexicalized items used to build it. For example, here is an English joke which uses substitution:

> What do you give an elephant that's exhausted? *Trunkquillizers.* [Webb, 1978]

In this joke, the word "trunk", which sounds like the syllable "tranq", is substituted into the valid English word "tranquillizer". The constructed word "trunkquillizer" is given a description, conveyed in the question part of the riddle, which is a combination of the meanings of "trunk" and "tranquillizer":

trunkquillizer: A tranquillizer for elephants.

So, there are several different tasks to be done in constructing a joke of this type:

1. A non-lexicalized word/phrase must be constructed.
2. A plausible meaning for that segment must be built.
3. A question-answer pair must be constructed, using the word/phrase and its meaning.

Of these, only the first is humour-specific, in that the mechanism (in this case, substitution) which builds the word/phrase must follow semantic/phonetic patterns that will result in a suitable core for a riddle. The second and third steps are general linguistic tasks. Given a sensible word/phrase, step two would construct a sensible meaning for that phrase; likewise, step three would construct a reasonable question-answer pair. In this model, step one is done by the schemata (section 4.3); step two is done by the small adequate description generator (section 4.4); and step three is done by the templates (section 4.5). Briefly,

1. **Schemata** constrain and assert relationships between lexical items and constructed items (including descriptions).
2. The **small adequate description generator**, given a sequence of lexemes, constructs a plausible description of the concept referred to by that sequence.
3. **Templates** translate instantiated relationships (between constructed items and descriptions) into question–answer pairs.

Each of these steps requires knowledge in order to be successful. Some of this knowledge is phonological, semantic and syntactic information about the words used to construct the riddle (section 4.1); some is information about what constitutes a plausible, adequate description of something (section 4.4); and some is knowledge about what forms a question-answer pair can take (section 4.5).

4.1 The lexicon

In order to construct jokes based on linguistic ambiguity, the system must have access to large amounts of linguistic information. In particular, it requires: *phonological information,* so that the phonological similarities or differences between two texts can be determined; *semantic information,* which can be used to evoke particular concepts or words in the listener; *syntactic information,* so that the words can be used grammatically in the final joke; and *surface form information,* so that similarities in the appearance of two texts (i.e. their spelling) can be determined.

The store of this information is the lexicon. The lexicon required for our needs is not humour-specific; that is, it contains no information put there for the purpose of creating jokes.

4.2 Small adequate descriptions

When we wish someone to think of a particular word, we often cue that word with a short (but not definitive) descriptive phrase. For example, we might use "an animal with a hump" to suggest "camel" – it is not usually necessary to give *all* the available information about a concept in order to bring it to the hearer's mind. These approximate descriptions are often used in riddles to sketch the meanings of punning phrases.

Here, a *small adequate description* (SAD) is the abstract, underlying structure for such a descriptive phrase. It lies somewhere on the boundary of syntax and semantics, using lexical properties to (approximately) describe some entity or set of entities. For example, the lexical entry for the word "lemon" may include the properties *class(lemon, citrus_fruit), adjective(lemon, sour), specifier(lemon, yellow), has(lemon, peel), has(lemon, seeds)*, etc. (where *lemon* is the lexeme). However, in order to adequately describe (not logically define) a lemon, it is only necessary to say "a yellow citrus fruit". So, <*lemon:* {*specifier(lemon, yellow), class(lemon, citrus_fruit)*}> is a SAD for the word "lemon". In our model, a SAD is always derived (by heuristic rules – see Section 4.4) from a tuple of lexeme(s), and is designed so that it will "describe" the same concept as a phrase constructed from those lexeme(s).

4.3 Schemata

A schema specifies the relationships between lexical items and 'constructed' concepts. It also relates lexemes to small adequate descriptions (SADs; see section 4.2). The schema itself does not *build* such descriptions — in our system, this is done by the SAD generator (see section 4.4) — but it does *constrain* them.

A schema consists of three types of information:

Lexical preconditions: These are various constraints on the lexical items and their properties which must be met before they can participate in the structures created by the schema.

SAD constraints: These specify the relationship(s) that must hold between the various items (either lexical items or constructed items) and the description that the SAD generator is to create; essentially, this is a specification to guide the SAD generator.

Descriptive conditions: These show the further relationship(s) which will hold between the constructed SAD and any other items involved.

For example, there is a substitution schema which could be glossed as follows:

Lexical preconditions: There is a compound nominal lexeme (*NPLex*) which is made up of the component lexemes [*LexA, LexB... LexN*] (in the order of their appearance in the surface form of *NPLex*). The spoken form of *LexA* is similar to the spoken form of another lexeme (*HomLex*). It is possible to construct a grammatically correct, non-lexicalized compound nominal (*NPWF*) from the sequence of lexemes [*HomLex, LexB... LexN*].

SAD constraints: Construct a SAD based on combining information from the pair of lexemes $\langle HomLex, NPLex \rangle$, treating *NPLex* as providing 'head' attributes and *HomLex* 'modifier' information.

Descriptive conditions: The SAD thus constructed describes the same concept as the constructed nominal *NPWF*.

Seen procedurally, this schema constructs a new compound nominal (e.g. "cereal killer") by substituting a similar–sounding word (e.g. "cereal") for the first word (e.g. "serial") in a lexicalized compound nominal (e.g. "serial killer"), and selects the entries for the similar-sounding word and the original compound nominal (e.g. "cereal" and "serial killer") as those from which to try to build a suitable SAD. With an appropriate template, it could generate:

> What do you call a murderer with fibre? *A cereal killer.* [JAPE]

4.4 SAD generator

In our model, the SAD generator contains a set of heuristic rules for extracting small adequate descriptions from lexical entries. Given a tuple of one or more lexemes, the SAD generator will 'guess' at an appropriate description. Its rules have to take account of the asymmetry between the 'head' of a phrase and its 'modifiers', and we have adopted the convention that the last item in the tuple will provide 'head' information. For example, given the tuple of lexemes $\langle aide, lemon\_ade \rangle$, the generator might come up with the following set of SADs (for 'lemon_aide'):

<lemon_aide: {*class(lemon_aide, drink), specifier(lemon_aide, helpful)*}>,
<lemon_aide: {*class(lemon_aide, assistant), specifier(lemon_aide, bitter)*}>.
That is, a "lemon aide" can be described either as "a helpful drink" or as "a bitter assistant". Note that the SAD generator can work equally well on either sensible lexeme tuples (e.g. $\langle fire, fighter\rangle$) or nonsense ones (e.g. $\langle aide, lemon\_ade\rangle$), and is therefore not humour-specific.

4.5 Templates

Templates transform a set of relationships between lexemes and descriptions into a suitable surface form for a punning riddle. Although we have selected forms suitable for question-answer punning riddles, these templates are not humour-specific; that is, they can also be used to construct non-joke question answer pairs, given a set of 'sensible' relationships.

A template consists of the following information, all interlinked by the use of logical variables:

Semantic relationships: These are constraints on the configurations of SADs and lexemes which the template can handle.

Sentence form: This is a piece of text containing some gaps, with variables indicating which of the items mentioned in the 'semantic relationships' are to be placed in which gaps. There are also details of the grammatical attributes that must belong to each of the items which fill those gaps.

For example, there is a English template as follows:

Semantic relationships: There must be an textual form *NPWF* and a SAD of the form < Lex : {*class(Lex, Class), specifier(Lex, Spec)*}> which describes the same concept (where *Lex* is the lexical form of *NPWF*, and is merely a place-marker if *NPWF* is a non-lexicalized textual form).

Sentence form: What do you call *NP1*? *NP2*.
where *NP2* is *NPWF* with an appropriate article (`a` or `an`) added and *NP1* is a noun phrase constructed from *Class* and *Spec*.

This template could be instantiated with *NPWF* = "low-comotive", *Class* = **train**, and *Spec* = **depressed**, producing the instantiated sentence form:

> What do you call a depressed train?
> *A low-comotive.*

In this way, a template provides the final form of a riddle specified in an instantiated schema.

5 Implementation of the English-language system

The model described above was implemented in a computer program called JAPE-2 (Joke Analysis and Production Engine 2) which generates punning riddles in English.[3] It has several distinct knowledge bases with which to accomplish this task:

- a combination of lexical resources, which contains humour–independent semantic (WordNet [Miller *et al.*, 1990]), syntactic (WordNet), and phonological (BEEP [Robinson, 1996] and a homonym list [Townsend and Antworth, 1993]) information about the words and common compound nominals;
- a set of thirteen schemata, all of which are specializations of the substitution, juxtaposition, and comparison mechanisms;
- a set of 21 templates suitable for the current lexical resources; and
- a SAD (small adequate description) generator, containing 10 rules suitable for use with our current lexical resources.

At the top level, JAPE-2 chooses a schema (or it could be specified by the user), and instantiates it. Instantiating a schema involves generating SADs, which is done by the SAD generator. The relationships asserted in the instantiated schema are then passed to a template, which generates a suitable surface form for the riddle. All of these stages require information from the lexicon. Please see Section 8.1 for a brief description of the evaluation of JAPE-2.

6 Types of pun found in Japanese

Although most languages have puns in some form or another, the commonness, variety and use of puns varies considerably from language to language. Of the three mechanisms described in Section 3, puns using comparison do not seem to exist as a genre in Japanese, but the other two mechanisms can be found.

Puns using juxtaposition are quite common in Japanese. However, the puns using juxtaposition tend to involve high as well as low level ambiguities. Moreover, puns of this type have few surface similarities, and are rarely, if ever, found in riddle form. In fact, none of the question–answer riddles in our corpus were juxtaposition puns.

Puns using substitution are also very common in Japanese, and are very similar in form to their English cousins. For example (translated into English):

> What do you call an actress who can make candies? Asaoka Glico.

Here, Asaoka Ruriko is a famous actress, and /guriko/, or "Glico", is a confectionery company. In this pun, /guriko/ is substituted for the similar sounding /ruriko/.

Because word-sense and pronunciation ambiguity are both relatively rare in Japanese, most Japanese puns use spelling ambiguity. In contrast to English, puns using perfect homonyms belonging to the same syntactic category are not very common, apparently because same-category homonyms are too common to be of interest.

[3] An early prototype, JAPE-1 – making no use of SADs – was presented in [Binsted and Ritchie, 1994].

The substitution of similiar-sounding words is the most common punning mechanism in Japanese punning riddles.

The domain for the Japanese version of our system, therefore, is Japanese punning riddles which use substitution of similar-sounding texts. We do not attempt to model comparison or juxtaposition riddles in Japanese — the former because they do not seem to exist in the language, and the latter because they use high level ambiguities and do not usually appear in riddle form.

7 Implementation of the Japanese-language system

BOKE (Bilingual jOKe Engine[4]) is an implementation in Japanese of the model described in section 4. BOKE differs from the English system, JAPE-2, in only a few ways. The lexicon is necessarily Japanese. Only substitution schemata are used (for reasons discussed in section 6). Also, phonological similarity is determined by the algorithm described in [Takizawa, 1993], rather than the more simple technique used in JAPE-2. Finally, the templates are in Japanese, but are a direct translation of the English templates used in JAPE-2. Otherwise, the Japanese and English implemented systems are the same.

BOKE has been hooked up to three very different Japanese lexicons: a tiny hand–built lexicon, for testing purposes; EDR [Yokoi, 1995], a very large general purpose lexicon; and a small lexicon derived automatically from a Japanese thesaurus [Institute, 1964].

In theory, an implementation of our model can use any general purpose lexicon, as long as it has enough semantic, syntactic and phonological information. EDR's specifications suggested that it would meet our needs; however, we had a great deal of difficulty adapting EDR for use with BOKE. Problems included: distinguishing true synonyms from mere spelling variations; deriving typical and/or definitional semantic relationships; and eliminating obscure or obsolete words. For these reasons, we did not use EDR as a lexicon for BOKE.

Instead, we developed a small hand–built lexicon, which contains words defined by volunteers. This lexicon was very useful during development; however, since its construction followed a somewhat ad-hoc methodology, we could not confidently claim it as an unbiased source of non-humour-specific lexical information for the sake of the evaluation.

As a compromise between the small lexicon hand–built for BOKE, and the large, unbiased (but not very useful) EDR, we also built a small lexicon derived from a Japanese thesaurus [Institute, 1964], and filtered using a small dictionary [Hadamitzky, 1992] as an approximation to a list of common words. The information was automatically extracted from the electronic form of the thesaurus, and so could be assumed to be non-humour-specific.

[4]In Japanese, a "boke" is the funny half of a funny-man/straight-man comedy team.

8 Evaluation

8.1 The English-language system

Here we summarize the evaluation of JAPE-2, which has been described elsewhere [Binsted *et al.*, 1997] in considerable detail.

To evaluate the quality of the punning riddles generated by JAPE-2, we asked 120 8-11 year-old children to act as 'joke judges'. Each child was given twenty texts to judge. The texts were randomly chosen from these four types: texts generated by JAPE-2; published human-generated punning riddles; nonsense non-jokes; and sensible non-jokes (i.e. pairs in which the answer is a non-humorous and accurate response to the question). For each text, the children were asked "Is this a joke?" [yes/no], "How funny is it?" [1-5], and "Have you heard it before?" [yes/no].

The key results were as follows:

- **Jokiness**: 81% of the human-generated texts were judged to be jokes, as were 60% of the JAPE-2 generated texts, 23% of the nonsense non-jokes, and 20% of the sensible non-jokes. All differences are significant except that between the two kinds of non-jokes.
- **Funniness**: Of those texts which were judged to be jokes, the human-generated jokes were given an average funniness score of 3.57, JAPE-2 generated texts 3.14, nonsense non-jokes 2.95, and sensible non-jokes 2.9. Again, all differences are significant except that between the two kinds of non-jokes.
- **Readability**: When texts containing words with low familiarity, imageability or concreteness scores (as recorded in the MRC Psycholinguistic Database [Wilson, 1987]) were removed from consideration, the average 'jokiness' of the remaining JAPE-2 texts was not significantly different from that of the remaining human-generated texts.

So, we found that computer-generated puns are recognizable as jokes, and (when corrected for vocabulary level) are just as joke-like as those published in children's joke books.

8.2 The Japanese-language system

The BOKE evaluation was similar to that of JAPE-2, although on a smaller scale. The BOKE output we tested included texts generated both from the thesaurus-derived lexicon and from the hand-built lexicon. The human-generated texts came from a published book of children's riddles [Konomi, 1996]. The sensible question-answer pairs were generated by a volunteer filling in BOKE templates. The nonsense non–joke question–answer pairs were generated by randomly selecting words from the thesaurus [Institute, 1964] to fill the slots in BOKE's sentence forms.

These texts were divided into twelve sets, randomized, and sent to twelve adult Japanese volunteers by email. For each text, the volunteer was asked to indicate whether or not the text was a joke (yes, no or don't know), if it was a pun (yes, no, or don't know), and if it was a riddle (yes, no, or don't know). They were also asked to indicate how funny it was, on a scale of one to five, and to note any comments they might have.

The results were as follows [HBT = BOKE texts using the hand-built lexicon; BT = BOKE texts using the thesaurus; HT = human–generated texts; NT = nonsense texts; ST = sensible texts]:

"Is it a joke?" There was no significant difference in how 'joke-like' the different types of text were judged to be. In fact, *none* of the texts (including HT) were judged to be jokes by more than one volunteer.

"Is it a pun?" HBT were significantly more 'pun-like' than any other kind of text. There was no significant difference between HT and BT, but both were significantly more pun-like than both NT and ST. There was no significant difference between how pun-like NT and ST were judged to be.

"Is it a riddle?" HT were significantly more 'riddle-like' than all other kinds of text. There were no other significant differences.

"How funny is it?" HT and HBT were significantly funnier than all other kinds of text. There was no significant difference between HT and HBT. BT were significantly funnier than NT. There were no other significant differences.

The results of the evaluation strongly suggest that BOKE does in fact generate puns in Japanese. BOKE-generated texts were significantly more pun-like than the non-joke texts, and the BOKE texts generated using the hand-built lexicon were significantly more pun-like than all other kinds of text, including the punning riddles from [Konomi, 1996].

Moreover, they suggest that the *lexicon* is the main factor determining the funniness of the puns generated by BOKE. The semantic relationships in the hand-build lexicon were richer than those given in the thesaurus, which only had 'synonym' and 'class' links. Also, all the words in the hand-built lexicon were in common usage, and included some celebrity names and modern slang; whereas the thesaurus-based lexicon included many obscure or obsolete words. We feel that these factors contributed significantly to the funniness, and general comprehensibility, of the texts generated by BOKE using the hand-built lexicon.

We were surprised to see that almost none of the texts, including the human–generated punning riddles, were judged to be jokes, even though many were thought to be riddles and/or puns. In English, punning riddles are generally considered to be a kind of joke. Our evaluation, however, suggests that this is not the case in Japanese. This comment was typical:

Jokes must include social wit or irony, so this questionnaire's play-on-words include no jokes, I think.

Another counter–intuitive result was that even those BOKE texts that were judged to be puns, were not, in general, judged to be 'riddles'. In English, a pun in question–answer form is generally considered to be a type of riddle. However, it seems that the translation we have been using for "riddle" (nazo-nazo), refers to a *children's* genre of question–answer texts, sometimes intended to be funny, sometimes more quiz-like. We suspect that the relatively high vocabulary level of the BOKE texts prevented them from being seen as nazo-nazo, as they would not, in general, be understood by children.

9 Conclusions

In this paper, we have presented a formal model of punning riddles, and two implementations of that model — one in English and one in Japanese. Evaluations of both systems indicate that texts generated in accordance with this model are recogizable as puns, and are comparable in quality to those generated by humans and published in children's joke books. The fact that one model successfully describes punning riddles in two very different languages suggests that the model captures the essential features of punning riddles.

References

[Binsted and Ritchie, 1994] Kim Binsted and Graeme Ritchie. An implemented model of punning riddles. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, USA, 1994. Available as Research Paper 690, Department of Artificial Intelligence, University of Edinburgh.

[Binsted *et al.*, 1997] Kim Binsted, Helen Pain, and Graeme Ritchie. Children's evaluation of computer-generated punning riddles. *Pragmatics and Cognition*, 5(2), 1997.

[Ertner, 1993] James D. Ertner. *Super Silly Animal Riddles.* Sterling Publishing Co. Inc., New York, 1993.

[Hadamitzky, 1992] Wolfgang Hadamitzky. *Langenscheidts Lehrbuch und Lexikon der jap. Schrft.* LANGENSCHEIDT-Verlag, Berlin, 1992. Online version entered by Helmut Goldenstein.

[Institute, 1964] National Language Research Institute, editor. *Word List by Semantic Principles, Source VI.* Shuei Shuppan, Japan, 1964.

[Konomi, 1996] Hikaru Konomi. *Nazo-nazo asobi 3 boukenpyokotan.* Akane Shobou, Japan, 1996.

[Miller *et al.*, 1990] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, Katherine Miller, and Randee Tengi. Five papers on WordNet. *International Journal of Lexicography*, 3(4), Winter 1990. Revised March 1993.

[Robinson, 1996] Tony Robinson. The British English example pronounciation dictionary. online version, 1996.

[Takizawa, 1993] Osamu Takizawa. Characteristics of some sorts of connotations in spoken language. *Journal of the Communications Research Laboratory*, pages 171 – 178, 1993.

[Townsend and Antworth, 1993] William Townsend and Evan Antworth. *Handbook of Homophones (online version).* 1993.

[Webb, 1978] Kaye Webb, editor. *The Crack-a-Joke Book.* Puffin, London, 1978.

[Wilson, 1987] Michael Wilson. The MRC psycholinguistic database. Technical report, Informatics Division, Science and Engineering Research Council, Rutherford Appleton Laboratory, Oxford, 1987.

[Yokoi, 1995] Toshio Yokoi. The EDR electronic dictionary. *Communications of the ACM*, 38(11):42–44, 1995.