

On a Possible Generative Approach to Structurally Ambiguous Humor

Dallin D. Oaks

Brigham Young University

Linguistics & English Language Dept.

4064 JFSB, BYU

Provo, UT 84602

dallin_oaks@byu.edu

Abstract

This paper outlines a generative approach for creating structurally ambiguous humor. The approach builds upon a lexical set that has been derived through a script associated with a given situation. Each of these lexical entries would also contain one or more specified SAPs (“structural ambiguity potentials”), which serve to match the lexical item to designated formulas for creating structural ambiguity. As part of the approach, an additional phonological and morphological component would serve to generate additional lexical forms with their own SAPs and related formulas. The paper also illustrates how the resulting structural ambiguities can then be systematically integrated into a humorous context.

Introduction

Among the range of humor types that deserve serious exploration in approaches to generating humor are those that are based around a structural ambiguity, that is, an ambiguity involving the way a particular structure is to be parsed. In English many of the jokes based on structural ambiguities involve syntactic structures whose potential for ambiguity is widely known and recognized among those who study the structure of the language. But although humor theorists recognize the role that structural ambiguity has in some humor, most would probably not see its underlying syntactic potentials within the language as a significant factor in generating humor. Yet I believe that the syntax of English presents so many identifiable options for structural ambiguity that those potentials could be integrated within a systematic approach towards generating some humor. Of course such an approach must interface with semantic and pragmatic mechanisms in order for the structurally ambiguous potential to be realized and to be

seen as humorous. But the development of various ontological apparatuses makes me think that at some point a generative approach to structurally ambiguous humor will not only be feasible but pose a useful resource to those working in computational approaches (for some examples of computational approaches to humor or theoretical models that can be related to such computational approaches, see Raskin, 1985; Attardo and Raskin, 1991; Ritchie, 2004; Attardo, 2008; Hempelmann, 2008; and Strapparava, Stock, and Mihalcea, 2011).

In this paper I will outline some mechanisms and strategies that a generative approach could employ as it works through a sort of decision tree format that could potentially generate a significant amount of humor based on structural ambiguities. Much of what I outline here is found in Oaks 2010, though this paper extends that earlier research.

Outlining a Generative Approach

Lexical SAPs and Related Formulas

The initial step in my approach would not begin with a structural ambiguity itself. The humor would be generated from the lexicon for one or more scripts associated with a particular event or setting (cf. Raskin 1985). The lexicon, as might be imagined, is a crucial component, so its structural dimensions deserve some attention here. Each lexical entry would not merely identify the so-called part of speech of a word but its syntactic behavior that is crucial to the development of structural ambiguities, what I will refer to here as its “structural ambiguity potential” (or SAP). Fortunately, some of the better existing lexicons probably already specify much of the important syntactic information that will be useful for the SAP component. But this information must be configured in the lexical entries in such a way that a computer could automatically

draw upon it to generate humor. Now I will elaborate further on how this might be done.

Within the SAP component of the lexical entries, we could use labels that tie the lexical item to a specific formula that could subsequently be applied. Among the specified information would be the homonymous or polysemous relationship a word has with another lexical category or subcategory. Thus for example, a word that may be both a count noun and a gradable adjective, or a word that can be both a count noun and a non-count noun would be noted in the SAP. This specification can then be matched with specific formulas that have been identified for such homonymous or polysemous words. In a two-volume work titled *Structural Ambiguity in English: An Applied Grammatical Inventory*, I have identified specific formulas for words belonging to more than one lexical category. While some further refinements are needed for the kind of lexical labeling that has been done, the identified formulas will go a long way towards generating well-formed syntax for both interpretations of specific homonymous and polysemous forms (Oaks 2010). Of course a subsequent semantic/pragmatic interface will be crucial, and this will be briefly outlined later in this paper.

Let's take a word like *novel*, which would be specified in its lexical entry as both a count noun and an adjective. With this designation it can immediately be directed towards an entire set of formulas that have been identified for creating a structural ambiguity based around that specific form. Among the larger list of formulas that can be drawn from Oaks 2010 for working with a word able to be interpreted as both a count noun and an adjective are the formulas given below. Note that these formulas involve syntactic contexts in which the obligatory presence of a determiner to accompany a singular count noun and thus signal its identity is somehow undermined or neutralized:

1. Use the noun/adjective word as a subject complement in a predicate structure after BE, BECOME, or REMAIN and the construction “a little.” Example: “It was a little novel.”

Note that noun/adjective words used in this formula must, by their adjective sense, not only be gradable but also have a semantic -1 designation,¹ meaning that they have a slightly negative quality. For example, although *safe* is a noun/adjective word, its adjective sense would normally resist a sentence like “It was a little safe.”

Note also that here and in the next formula (formula #2), the use of “a little” is unnecessary for noun/adjective words that are non-count by their noun interpretation, such as “It was fat.” Furthermore, while this formula (as well as formula #2) works well

with noun/adjective words, it should not be used with noun phrase/adjective words such as *ajar* and *adrift*.

2. Use the noun/adjective word as a subject complement/direct object after GET and the construction “a little.” Example: “We got a little novel.”
3. Use the noun/adjective word as a premodifier to a head noun in a noun phrase structure. Example: “A novel opinion”
4. Use the noun/adjective word after a suffix/contraction -‘s that is attached to a noun. Example: “The publisher’s novel.”
5. Form a simple noun phrase structure consisting of the noun/adjective word as the head element of a compound noun or after a noun premodifier. Use this noun phrase as the subject of an SVA clause (Subject + BE + Adverbial of Place). Then change this structure into the corresponding yes/no question. Through this formula the ambiguous “Was the adventure novel only in his head?” could come from either “The adventure novel was only in his head” or “The adventure was novel only in his head.”

This partial set of formulas here is provided only to give an idea of the ways that a lexical entry that marks a homonymous or homophonous potential could be matched with identifiable structures that could result in structural ambiguities.

The SAP would also be identified within the lexical entries of other lexical categories as well. For example, verbs may have their SAP marked not only with such information as whether the verbs are homonymous or homophonous with other lexical categories such as we see with *dance* (a noun/verb word), but also to indicate other significant features of their SAPs, including what kinds of clause types those verbs may create. For example, a verb like BRING can generate both an SVO and SVOO clause, allowing for a structural ambiguity like “The boy brought the cowboy boots.” And FIND can generate both an SVOO and SVOC clause, allowing for a structural ambiguity as in “The host found her an interesting conversationalist.” Such verbs can be specified in such a way as to be matched with identifiable formulas for creating a variety of structural ambiguities.

The verbs to be manipulated enter into the ambiguity approach not only through the initial stage that uses scripts to produce a set of words for generating humor, but also in some cases as a result of the SAP of specific words that have already been collected through the initial scripts stage. For example, compound count nouns (or count nouns with a preceding modifier) when pluralized, such as “library books,” would be specified to draw forward verbs

¹ I am grateful to Paul Baltes for pointing out this particular convention for marking negative connotations in lexical entries.

such as BRING, MAKE, BUY, and SELL that are specified in their SAP component as allowing both SVO and SVOO clause structures.

Beyond the SAPs that can be specified in lexical entries for open classes such as nouns, verbs, adjectives, and adverbs, we can also identify the SAPs for the lexical entries of words belonging to the closed classes such as pronouns, conjunctions, and prepositions. In the case of the words from the closed classes, the starting point will not be so much an identification of words through a script-based listing of terms but rather through relationships and circumstances that will suggest specific closed class words. For example, a sequencing of events could present the preposition *before*, whose SAP can involve a ‘space vs time’ interpretation. Furthermore, contexts requiring descriptions and limitations can be matched with known formulas for modifier ambiguities.

Now let’s back up and consider how the kinds of specification we have identified for the lexical entries can work in a larger generative process. In Oaks 2010, we can see a methodology that uses several stages. First, as we have seen, we have the initial identification of words around which a structural ambiguity can be built. These were identified through a semantic/pragmatic script component. Thus if we wanted to create a joke about air travel or a common circumstance accompanying air travel, we might access a script that would yield words such as *pilot, airplane, seat, radio, terminal, control tower, flight attendant, security, luggage, uniform, cockpit, fly, landing, departure gate, aboard, purchase, jet lag, deplane, Federal Aviation Regulation, exit, stranger*, etc. These words would have certain SAPs already marked in their lexical entry. For example, *pilot, seat, radio, fly, purchase*, and *exit*, are noun/verb words that can be matched with their own syntactic formulas for structural ambiguity. But let’s continue to focus here on noun/adjective words like *terminal, uniform*, and *stranger*, which form part of the air travel word set and which, as we have seen, can be matched with formulas that have been designated for use with noun/adjective words, and, as we shall see, can be matched with a humorous context.

Before moving on, it will be useful to consider how the approach identified here can develop additional forms through some specific alterations to the forms or pronunciations of the words that have already been presented. These alterations have the goal of increasing the number of relevant words around which word plays may be produced. This alteration stage, among other elements, has a phonological as well as a morphological component. The phonological component will explore certain transformative processes like assimilation, common phonological substitutions, and word boundary alterations that can apply to words and word phrases. Such mechanisms within my approach could take a word like

deplane and specify a shared identity between *deplane* and “the plane,” specifying the oral form as a verb/noun phrase form. Similarly, a word boundary alteration would also provide for *aboard* to be a noun phrase/adjective form (*a board* and *aboard*). Next, we apply a morphological mechanism that includes such things as the ability to produce additional forms of words that take into account such common processes as clipping, blending, abbreviations, and acronymy. From these processes we would get the additional forms like *plane* from *airplane* and *FAR* from *Federal Aviation Regulation*.

The use of these phonological and morphological mechanisms to produce additional forms and identities is not trivial. It allows for additional SAPs that can result in further structural ambiguities. For example, unlike *airplane*, which can only be a noun, *plane* can be homophonous as both a noun or an adjective, allowing for an ambiguity like “It was a little plain/plane.” And when the spoken form *aboard* acquires the additional identity as “a board,” the spoken form may be effectively combined with a verb like BRING, which can be both monotransitive and ditransitive to create a structural ambiguity such as “He brought her aboard/a board.” Moreover, by altering “Federal Aviation Regulation” to its established acronym, we can get a form that may be a proper noun/adjective/adverb form. The alteration stage also adds to the script-derived lexicon by locating synonymous forms with alternative SAPs. The verb CREATE, for example, allows for only an SVO structure, but its synonym MAKE allows for SVO, SVOO, and SVOC structures. Thus we can get additional possibilities for structural ambiguity. After doing this we end up with a larger list of lexical forms and phrases that have been specified with SAPs, or in the case of forms like *deplane* and *aboard*, orally-based SAPs that were not previously found in our list of words identified from the script. These additional forms can then also be applied to a humorous context.

Applying Structural Ambiguities to a Humorous Context

In an earlier version of my work I showed how an advertiser or marketer could use a knowledge of lexical categories and structural ambiguity formulas to create catchy lines for slogans, business logos, or advertisements. This of course relied on a human to carry out the process and make the necessary connections and then to act as a filter in applying the appropriate semantic/pragmatic context in order to make the generated output humorous or clever. That approach is potentially effective for the kind of application performed by advertisers and marketers, but

it is quite different from what is needed for a computational system. The question then arises here about whether my approach could apply to a more automatic, non-human system for generating humor. Can we get from the specified lexical entries and structural formulas to actual joke output? This step, as might be expected, is rather tricky. But I was able to come up with a type of joke that my system seems able to generate, though it relies on advances in ontological semantics and pragmatics. Consider the following joke framework, which is based on the first formula for noun/adjective words that was identified earlier:

Why did the _____ [sentient being that is primarily associated with the script of the noun sense of the noun/adjective word] **dislike the _____?** [singular noun that is roughly synonymous with the noun sense of the noun/adjective word].

Because _____ [singular pronoun that matches the gender of the noun sense of the noun/adjective word] **was a little _____.** [noun/adjective word]

Thus we could have riddles like the following ones using the noun/adjective words *novel* and *grave*:

Why did the reader dislike the book? Because it was a little novel.

Why did the mortician dislike the burial site? Because it was a little grave.

Now let's return to our list of noun/adjective words identified through the air travel script and augmented through phonological and morphological alteration strategies: *uniform*, *stranger*, *plane* (plain), *terminal*, and *FAR*. All of these will be run through our joke framework above except the word *terminal*, since as we noted, the joke framework is based on the first formula, which requires that the adjective sense of the noun/adjective words be gradable, and the adjective *terminal* is not usually gradable. The remaining four words could then generate the following jokes, with perhaps only the last one being genuinely problematic:

For *uniform*: **Why did the flight attendant dislike the outfit? Because it was a little uniform.**

For *plane* (plain): **Why did the pilot dislike his aircraft? Because it was a little plane.**

For *stranger*: **Why did the passenger dislike the man in the adjoining seat? Because he was a little stranger.**

? For *FAR* (Federal Aviation Regulation): **Why did the lawyer dislike the new government regulation? Because it was a little FAR.**

The kind of interface used in the joke framework would of course need to be refined some, since it doesn't work perfectly, but my examples give a rough idea of how we could take an approach that is heavily grounded in syntactic potentials within the language and integrate it with a semantic/pragmatic interface to generate humor.

Conclusion

The preceding jokes rely on only one of my formulas growing out of just one of the SAPs identified in my lexical entries, and it could be easy to wonder whether this joke potential is a fluke of sorts. But I think with additional work and ingenuity we could also exploit other identified formulas to develop additional joke strategies that could operate productively to generate structurally ambiguous jokes. On a recent trip a flight attendant gave me a packaged snack that bore the label "Plane Cookie." I reflected on the fact that this product name could be developed through the word alteration strategy and the third of my noun/adjective formulas that I had already identified. The third noun/adjective formula was also used by the playwright, Dennis Potter, who said, "I did not fully understand the dread term 'terminal illness' until I saw Heathrow Airport for myself." The productive nature of the third noun/adjective formula to generate word plays makes me think that we may be able to develop a joke framework interface that would generate humor for that type of syntactic formula as well. More significantly, through the use of lexical entries that specify the SAPs in ways that combine with syntactic formulas for creating structurally ambiguous word plays, we may be able to devise a variety of joke frameworks that generate humor that is based on structural ambiguities.

References

- Attardo, S. 2008. A Primer for the Linguistics of Humor. In *The Primer of Humor Research*, ed. V. Raskin, 101-55. Berlin: Mouton de Gruyter.
- Attardo, S. and Raskin, V. 1991. Script Theory Revis(it)ed: Joke Similarity and Joke Representation Model. *Humor: International Journal of Humor Research*. 4 (3/4): 293-347.
- Hempelmann, C. F. 2008. Computational Humor: Beyond the Pun? In *The Primer of Humor Research*, ed. V. Raskin, 333-60. Berlin: Mouton de Gruyter.
- Oaks, D. D. 2010. *Structural Ambiguity in English: An Applied Grammatical Inventory*. 2 vols. London & New York: Continuum.
- Raskin, V. 1985. *Semantic Mechanisms of Humor*. Dordrecht: D. Reidel Publishing Company.
- Ritchie, G. 2004. *The Linguistic Analysis of Jokes*. London & New York: Routledge.

Strapparava, C.; Stock, O.; and Mihalcea, R. 2011. Computational Humour. In *Emotion-Oriented Systems*, ed. P. Petta, C. Pelachaud, and R. Cowie, 609-34. Berlin: Springer-Verlag.

www.all-acronyms.com

www.englishclub.com/english-for-work/airline-vocabulary.htm