Phonological Pun-derstanding

Aaron Jaech and **Rik Koncel-Kedziorski** and **Mari Ostendorf**
University of Washington
Seattle, WA
{ajaech,kedzior,ostendor}@uw.edu

Abstract

Many puns create humor through the relationship between a pun and its phonologically similar target. For example, in "Don't take geologists for granite" the word "granite" is a pun with the target "granted". The recovery of the target in the mind of the listener is essential to the success of the pun. This work introduces a new model for automatic target recovery and provides the first empirical test for this task. The model draws upon techniques for automatic speech recognition using weighted finite-state transducers, and leverages automatically learned phone edit probabilities that give insight into how people perceive sounds and into what makes a good pun. The model is evaluated on a small corpus where it is able to automatically recover a large fraction of the pun targets.

1 Introduction

From the high culture of Shakespeare's plays (Tanaka, 1992), to the depths of the YouTube comments section, from advertising slogans (Keller, 2009) to conversations with nerdy parents, puns are a versatile rhetorical device and their understanding is essential to any comprehensive approach to computational humor. Humor has been described as "one of the most interesting and puzzling research areas in the field of natural language understanding" (Yang et al., 2015). Puns, in particular, offer an interesting subject for study since their humor derives from wordplay and double-meaning.

An important class of puns, known as paronomasic puns, are those where one entity, the pun, is phonologically similar to another, the target (Joseph, 2008). Consider an example from Crosbie (1977):

> "Sign by gate to nudist colony: Come in. We are Never Clothed."

Here, "clothed" is the pun and "closed" is the target. Paronomasic puns are distinguished from homographic puns such as

> "Two silkworms had a race. They ended up in a tie."

which puns on the two definitions of the word "tie". When the pun and target are homophonic this is called a *perfect* pun, and when nearly homophonic an *imperfect* pun (Zwicky and Zwicky, 1986) (or a *heterophonic* pun (Hempelmann, 2003)). The focus of this work is to propose and evaluate a model for target recovery of both perfect and imperfect paronomasic puns, assuming that the location of the pun word or word sequence.

Ritchie (2005) classifies puns in terms of whether they are *self-contained*, i.e., based on general knowledge and humorous in a variety of circumstances, or *contextually integrated*, i.e. relying on a specific context such as a visual context, knowledge of a recent event or discussion. Many puns of this type are associated with cartoons or images, e.g. a cartoon with pies and cakes in the street having the caption

> "The streets were oddly desserted"

(desserted/deserted). Contextually integrated puns lose their humor out of context because the pun is difficult to detect. However, target recovery is often still possible, and thus this distinction does not play a major role in the current study.

If a listener fails to recover the target of the pun then the statement fails in its humor. The two chief

Proceedings of NAACL-HLT 2016, pages 654–663,
San Diego, California, June 12-17, 2016.

clues that the listener must rely on to perform target recovery are the phonetic information and language context. This is analogous to the way in which someone listening to speech uses the acoustic information and language context to recover a sequence of words from an audio source. In the words of Hempelmann (2003), "the recovery of the target in heterophonic puns is just a specific case of the complex task of hearing." The similarity between hearing and target recovery suggests the use of methods from automatic speech recognition in building a model for automatic target recovery.

The goal of this paper is to develop and test a computational model for target recovery in puns. Sentences with the position of the pun marked are given as input and the model must output the target word sequence. As the focus is on paronomasic puns, the relationship between the pun and the target is primarily phonological, but surrounding language context is also important for recovering the target. This work has applications in natural language understanding of texts that contain humor. Furthermore, the insights gained from our model are useful for improving pun generation in computational humor systems.

2 Prior Work

Zwicky and Zwicky (1986) provided an analysis of the properties of paronomasic puns, especially with regard to the markedness of phonological segments. They assembled a corpus of 2140 instances of segmental relationships from imperfect pun/target pairs for their analysis. By counting how many times each phoneme was used in a pun or target, the authors observe a behavior they refer to as ousting, a strong asymmetry in phoneme substitution likelihood. For instance, punners will rarely replace a 'T' phoneme (IPA t) in the target word with 'TH' (θ) in the pun, but regularly replace 'TH' with 'T.' An example of such a pun is

"I lost my temper in a fit of whiskey"

(fit/fifth)). Because of this asymmetry, we say that 'T' ousts 'TH.' Zwicky and Zwicky correlate the ousting behavior evident in their pun data with the phonological notion of *markedness*. Markedness can be defined (if oversimplified) as "the tendency for phonetic terms to be pronounced in a simple, natural way" with regard to physiological, acoustic, and perceptual factors (*marked* segments are more complex) (Anderson and Lightfoot, 2002). They conclude that marked segments tend to oust unmarked segments (e.g. voiced stops oust their voiceless counterparts).

This is followed-up by Sobkowiak (1991), whose manual alignment of the phoneme sequences of 3,850 pun/target pairs allows for a more careful study of the ousting behavior. This corpus, where whole sequences of phonemes are aligned between puns and targets, is a much richer resource for analyzing ousting than the segment-only data used in (Zwicky and Zwicky, 1986). Sobkowiak's improved data provides evidence against the conclusions of Zwicky and Zwicky (1986). Rather, "it seems that it is not the case that 'marked ousts unmarked' in paronomasic puns." Sobkowiak goes on to show that puns more frequently involve changes to vowels than consonants, noting that their information load (i.e. contribution to target recoverability) is lighter. Our data corroborates Sobkowiak's claims regarding the role of markedness in punning as well as the mutability of vowels, and provides more details about the specific nature of substitutions in a large corpus of puns.

Building off of Sobkowiak's work, Hempelmann (2003) studies target recoverability, arguing that a good model for target recovery provides necessary groundwork for effective automatic pun generation. He proposes a preliminary phonetic edit cost table to be one part of a scoring system. The model is based on the phoneme edit counts from Sobkowiak (1991) with an ad-hoc formula for transforming the counts into substitution costs. However, Hempelmann makes no effort to empirically test his model at the recovery task. The model uses a subset of 1,182 puns from the 3,850 identified by Sobkowiak. This subset is the data used for training our phonetic edit models and we use Hempelmann's cost function as a baseline.

The task of automatic target recovery of paronomasic puns has not been previously attempted. Recently, Miller and Gurevych (2015) studied methods for automatic understanding of homographic puns using methods from word-sense disambiguation. Paronomasia is intentionally excluded from their data.

3 Target Recovery

Following the convention of Miller and Gurevych (2015), we assume that the position of the pun in the input sentence is known. The target recovery task is to identify the pun target given the pun and its left and right word contexts.

3.1 Model

The model has three parts: a phonetic edit model, a phonetic lexicon and a language model. The recovered target T^* is the word (or words) with the maximum probability given the pun P according to

$$T^* = \underset{T}{\operatorname{argmax}}\, p(T|P) = \underset{T}{\operatorname{argmax}}\, p(P|T)p(T),$$

where $p(P|T)$ is the phonetic edit model and $p(T)$ is the language model. The factorization into a language model and a phonetic edit model is similar to the classic approach to automatic speech recognition.

The implementation of the model uses weighted finite-state transducers (WFSTs) which have been adopted as a useful structure for speech decoding due to their ability to efficiently represent each of the relevant knowledge sources, i.e. phonetic information, phonetic lexicon and language model, in a single framework (Mohri et al., 2002; Hori et al., 2007). Finite-state transducers are finite-state machines with an input and an output tape. We will use WFSTs for our pun target recovery model. Each of the phonetic edit model (PEM), phonetic lexicon (L) and language model (LM) can be represented as WFSTs, which are joined together by applying the composition operation. The weights on the PEM and LM are negative log-likelihoods, and the lexicon has no weighting. Each of these models will be explained in further detail below. The target is given by the shortest path in the WFST:

$$(\mathrm{LC} \oplus \mathrm{P} \circ \mathrm{L}^{-1} \circ \mathrm{PEM} \circ \mathrm{L} \oplus \mathrm{RC}) \circ \mathrm{LM},$$

where P is the pun and LC and RC are the left and right word contexts. The symbols $^{-1}$, $\circ$ and $\oplus$ denote the inverse, composition, and concatenation operations respectively.

The sequence of operations $\mathrm{P} \circ \mathrm{L}^{-1} \circ \mathrm{PEM} \circ \mathrm{L}$ converts a pun to its phonetic form, expands it to a lattice based on phonetic confusions, and then converts the phone lattice to a lattice of possible target words. By concatenating the left and right word contexts and composing with the language model, each path through the WFST is a target word sequence with a weight equal to the combined phonetic edit and language model scores. The ability to handle multi-word puns and/or targets (e.g., the word sequence "no bell" can be matched to "Nobel", using an example from (Yang et al., 2015)) is made possible because the lexicon WFST L allows multiword sequences.

Since the scores are negative log likelihoods, the target hypothesis is just the shortest path in the WFST. We score the model based on its accuracy at identifying the target which must be an exact match, ignoring punctuation. We disallow the possibility that the pun is hypothesized as a target, i.e. homographic puns, in order to focus on the class of puns whose relationship with their targets is primarily phonological. The OpenFst library is used to perform all of the WFST operations (Allauzen et al., 2007).

3.2 Phonetic Edit Model

The purpose of the phonetic edit model is to estimate the probability of the pun phoneme sequence given a candidate target phoneme sequence. We prefer to learn a model from the data rather than adopt an existing model that relies on phonetic features and edit costs that were derived by hand (Kondrak, 2000). As shown by Ristad and Yianilos (1998), a memoryless WFST model can learn a probability distribution over edit operations with a principled objective, namely, to maximize the likelihood of the source/target sequences in the training data. In our case, the training data is pun/target phoneme sequences. Memoryless, in this context, refers to the fact that the model is not conditioning on previous symbols, i.e. there is only a single state in the WFST. The model assumes that the phoneme sequence of the pun is generated through the stochastic application of insertions, deletions, and substitutions to the target phoneme sequence. During learning, the model estimates the function $p(y|x)$ where y is a phoneme from the pun and x is a phoneme from the target. When $x = \epsilon$ this is an insertion of y and when $y = \epsilon$ it is a deletion of x.

Training uses the expectation-maximization algorithm following the equations given by Oncina and Sebban (2005) to estimate the conditional probability distribution instead of the joint one as originally derived by Ristad and Yianilos. The model is trained to maximize the probability of the pun targets given their sources subject to the constraint that the model encode a probability distribution over all possible string pairs. In the expectation step, puns are aligned with their targets given the current model. Then, the maximization step re-estimates the edit probabilities given the current alignment. Edit probabilities are initialized by giving a high probability to keeping the same phoneme and uniform small probabilities to all possible edits. (We initialized by giving ten times the probability to preserving the same phoneme as to any possible edit operation.) Following the convention of Sobkowiak (1991), vowels can not align with consonants and vice-versa.

The training data consists of the 1,182 target pun pairs taken from Appendix E of Hempelmann (Hempelmann, 2003). These are a subset of the puns that Sobkowiak took mostly from Crosbie's "Dictionary of Puns" and analyzed in his work.

3.3 Phonetic Lexicon

The lexicon models the pronunciation of each word in the vocabulary. Pronunciations come from the CMU pronunciation dictionary (Weide, 1998). This dictionary has an inventory of 39 phonemes. If a pun is not in the vocabulary of the dictionary, for example if it is not a word, then its pronunciation is generated automatically using the LOGIOS lexicon tool.[1] The same is not true for the targets, since they are unknown beforehand. Thus, when the lexicon is used to map puns to phonemes the vocabulary size is essentially unlimited. But, when it is used to map the phoneme lattice into a word lattice of potential targets then the fixed vocabulary from the language model is used.

The CMU dictionary includes multiple pronunciations for some words. All pronunciations are used with unweighted parallel paths. The version of the dictionary used here includes stress markers and syllable boundaries (Bartlett et al., 2009). In the simplest version of our model, this information is ignored in order to reduce the number of learned parameters in the PEM.

After composing with the phonetic edit model and the lexicon, we do a conservative pruning of the WFST to remove highly improbable word sequences based on the phonetic score and run epsilon removal on the resulting lattice. This reduces the memory footprint and allows use of a larger language model.

3.4 Language Model

A 230 million word corpus was formed from comments obtained from Reddit, an online discussion forum. These comments were collected from a wide variety of forums, known as subreddits. Reddit contributors tend to use a casual conversational style that is a good match for the language used in common puns. All of the text data from Reddit was tokenized using the NLTK tokenizer (Bird et al., 2009). The tokenizer splits contractions into two tokens but we kept these as a single token to match the pronunciation dictionary. We remove case information. Punctuation is removed when evaluating the correctness of the hypothesized targets, but punctuation symbols are included in the language model. It provides a useful context break in punning riddles where the pun/target typically follows a question mark or other punctuation.

The vocabulary is set by intersecting the vocabulary from the CMU pronunciation dictionary with the set of tokens that occur at least 30 times in the language model training data. This gives us a 36,175 word vocabulary. As Sobkowiak (1991) observed, the target tends to have a much higher unigram probability than the pun. This means that the vocabulary size need not be too large to cover most of the targets.

The language model is a trigram model with modified Kneser-Ney smoothing (Chen and Goodman, 1999). Entropy pruning is used to reduce the size of the language model (Stolcke, 2000). It is important to perform the determinization and minimization operations on the LM after converting it into the FST representation, in order to reduce the size of the model (Mohri et al., 2008). Because we are using a trigram model, only two words of context are needed on each side of the pun.

[1] http://www.speech.cs.cmu.edu/tools/lextool.html

3.5 Extending the Phoneme Edit Model with Syllable Structure and Stress

For a listener to recognize the phonological distinctness of the pun and the target, they should preferably differ in a perceptually salient position such as a stressed syllable. In particular, we expect that puns would take advantage of the increased acoustic energy in the onset and nucleus of a stressed syllable and utilize these positions for phoneme changes.

To analyze this effect we used a syllabified version of the CMU pronunciation dictionary (Bartlett et al., 2009). We took the 1-best phonetic alignment of the training data and split it according to the syllable boundaries of the pun. Then we computed the probability of a phoneme change according to the position in the syllable and the stress. Consonants in the onset of a stressed syllable have a 40.3% probability of changing between the target and pun. The nucleus of a stressed syllable of has a 36.8% probability of substitution. This is compared to a 31.8% probability of substitution for phonemes in unstressed syllables and coda positions.

To incorporate this into an extension of the phonetic edit model, we created a three state model. There is one default state and two special states for the stressed syllable onset and nucleus respectively. The phoneme edit probabilities $p(y|x)$ were scaled according to the state s and renormalized so that $p(y|x, s)$ is a valid probability distribution. When a substitution does occur, we assume that the choice of target phoneme is independent of the syllable position and stress. The net effect of the syllable extension to the PEM is to encourage substitution of onsets and nuclei of stressed syllables and discourage it otherwise.

4 Experiments

4.1 Data

We collected 75 puns from various joke websites such as Tumblr, Reddit, and Twitter and soliciting examples from friends and colleagues.[2] These were collected without reference to the sources used by Sobkowiak to assemble the puns used in building the phonetic edit model. This data was used for test data only and is completely separate from the training data used by the language model and the phonetic edit model. (It would be nice to have used some of the 1,182 puns from Sobkowiak for test data but only the isolated pun/target pairs were provided without the necessary word contexts.) Pun locations were marked in each sentence as the minimal set of words that change between the pun and the target.

Note that the phonetic edit model is trained on exclusively imperfect puns but it is tested on both perfect and imperfect puns (24 perfect and 51 imperfect). This creates a mismatch between the training data and the test data. Target recovery is harder on imperfect puns but having a mix of both types better reflects what is commonly found in the wild.

4.2 Baseline Model

As a baseline model we replace our phonetic edit model with the cost function proposed by Hempelmann, which we replicated based on details given in Appendix G of his thesis (Hempelmann, 2003). This cost function, which Hempelmann refers to as "preliminary," is the only published phonetic edit model for paronomasic puns. The cost table is based on the phoneme pair alignment counts from the 1,182 training pairs that were aligned by hand. The phoneme alignment counts are converted to costs by using simple ad-hoc equations. Vowel pair counts are transformed to costs using $cost = 0.3 - 0.3 * count/161$ and other pairs use $cost = count^{-0.6}$. Our model improves upon the baseline by avoiding heuristic transformations. Since the phoneme symbol set used by Hempelmann (based on (Sobkowiak, 1991)) differs from that used in the CMU dictionary, the Hempelmann costs are mapped to match the CMU inventory.

4.3 Results

We report the performance of our model in Table 1 using accuracy and mean reciprocal rank as metrics. If the correct target did not appear in our n-best list then we use a value of zero for its reciprocal rank. Ties are broken randomly. The baseline uses Hempelmann's phonetic cost model plus the LM, and we include two ablation models that use just the LM or just the PEM. The other two models use the LM with either the memoryless PEM or the PEM with the syllable extension.

[2] Data available at http://ssli.ee.washington.edu/data/puns.

Model	Accuracy			MRR
	Perfect	Imperfect	Overall	
LM Only	13.0%	7.7%	9.3%	0.127
PEM Only	43.5%	9.6%	20.0%	0.282
LM + Hempelmann	47.8%	7.7%	29.3%	0.389
LM + PEM	73.9%	65.4%	68.0%	0.729
LM + Syll. PEM	73.9%	65.4%	68.0%	0.733

Table 1: Accuracy and mean reciprocal rank (MRR) for target recovery

Using the language model only gives poor performance. It is only able to recover the target when the target happens to be an idiomatic expression. The PEM-only model does significantly better than using the LM only, highlighting the importance of phonetics in paronomasic puns. In the full system, Hempelmann's cost matrix does not fare well compared to the PEM model. The Hempelmann cost matrix does a poor job of separating likely targets from the rest of the vocabulary. Thus, many times the true target is pruned before the application of the language model.

The LM + PEM model recovers the target more than two-thirds of the time and has a mean reciprocal rank of 0.729. When using the syllable extension to the PEM, the results agree on the rank of the target for all but five puns. For those five, the model with the syllable extension improves the rank compared to the basic PEM. Two puns from that set of five are

> "If you've seen one shopping center you've seen a mall"
>
> "How does Moses make his tea? Hebrews it."

(a mall/'em all and Hebrews/he brews, respectively). The hypothesized targets "immoral" and "he abuse" outrank the true target for these puns in the basic PEM model but not in the syllable one because they change more phonemes in unstressed syllables.

Perfect puns are easier to recover than imperfect ones. The LM + PEM model does well on both perfect and imperfect puns. As to be expected, the PEM only model does very poorly on imperfect puns and the LM only model does equally poorly on both perfect and imperfect.

Table 2 shows the top ranked hypothesis for a sample pun using the LM + PEM model, where the cost in this table is the negative log-likelihood. In this case, the top ranked hypothesis was correct. The second highest ranked hypothesis is a misspelling of the target that is common enough for the language model to also give it a high score.

An example where the model makes a mistake is:

> "A Freudian slip is where you say one thing but mean your mother"

The pattern of "one thing . . . another" is common in English but, in this case, the target "another" is too far away from "one thing" for the relationship to be captured by the tri-gram language model.

5 Analysis

A consequence of using a stochastic model for phonetic edit costs is that there is a non-zero edit cost between a phoneme and itself and that cost is different depending on the phoneme. This highlights the fact that we model the edit (or transformation) probabilities of the pun/target corpus rather than phonological similarity (which would be a symmetric cost function). The analysis below shows that the edit model is in fact capturing more than simple phonological similarity.

5.1 Phoneme Edit Probabilities

Figures 1 and 2 show the probability of observing a source phoneme (i.e. a phoneme appearing in the pun) given each target phoneme for the vowel and consonant pairs respectively. The numbers are to be interpreted as percentages and values less than 1 are not shown. The '.' symbol is used for epsilon transitions and indicates segment insertions and deletions.

Vowels. Regarding vowels, our data corroborates some of the findings of Sobkowiak (1991). The lower numbers along the diagonal in Figure 1 relative to Figure 2 indicates the violability of vowels (39%) relative to consonants (34%) in paronomasic puns. Our data also confirms that, where puns are concerned, marked segments do not necessarily

Rank	Cost	Hypothesis
1	21.0	. . . ONLY GOT **MYSELF** TO BLAME
2	23.2	. . . ONLY GOT **MY SELF** TO BLAME
3	24.2	. . . ONLY GOT **A MYSELF** TO BLAME
4	24.2	. . . ONLY GOT **MY YOURSELF** TO BLAME
Source		. . . ONLY GOT **MY SHELF** TO BLAME

Table 2: Top ranked target LM + PEM hypotheses for the pun "A book fell on my head. I've only got my shelf to blame."

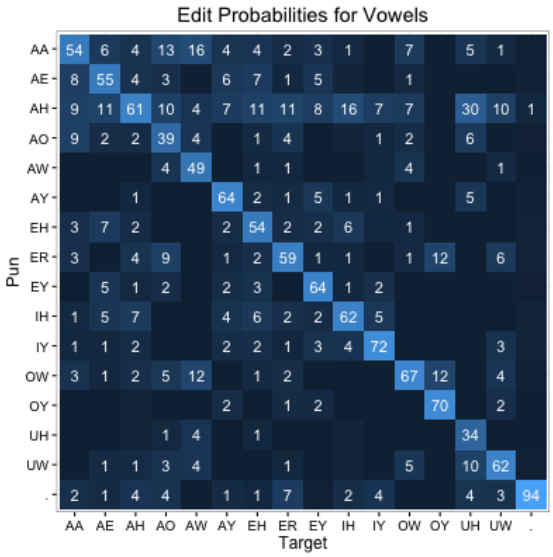

Figure 1: Edit probabilities for vowels based on the LM + PEM model.

oust unmarked. According to Zwicky and Zwicky (1986), if "marked ousts unmarked" we would expect to see that tense vowels oust lax vowels. Rather, what we find in this data is that the majority of vowels and diphthongs are ousted by AH (ə), widely considered an unmarked vowel. Puns demonstrating this phenomenon include

"The pun is mightier than the sword"

"He's an honest geologist, you can trust what he sediment"

(pun/pen and sediment/said he meant, respectively). We hypothesize that the low cost for substituting AH for other vowels is due in part to the fact paronomasic puns originally were a spoken phenomenon, and so the substitution possibilities for a given target vowel depend significantly on the variety of realizations of that vowel in speech. It is well attested cross-linguistically that vowels undergo reduction in unstressed positions (Crosswhite, 2004). In English running speech, this reduced form most closely resembles AH (Burzio, 2007). The data presented here suggests that punners take advantage of the commonality of running speech vowel reduction when considering target recoverability, resulting in the availability of AH as a replacement for most target vowels. Our model captures this fact by assigning low cost to such a substitution.

Our data provides other insights into the nature of vowel substitutions in puns. For instance, we see that IY (i) is less likely to change in a pun than other monophthongs, indicating a significant perceptual distance between IY and its neighbors. Most likely to change are the vowels AO (ɔ) and UH (ʊ). The violability of AO is likely due to the AO/AA merger present in many American English Dialects (Labov et al., 2006), and in fact we see targets with AO frequently mapping to puns with AA. The mutability of UH is also interesting: while it, like other monophthongs, is ousted by AH, it is also ousted by a closely articulated marked counterpart, UW (u). This seems to be the sole example in our data of marked vowels ousting an unmarked vowel to a significant degree.

Another interesting feature of this data is the substitutability of ER (ɝ) and OY (ɔɪ), as in puns like

"The British used to dress their sandwiches with earl and vinegar"

"In cooking class this week we're loining how to prepare tuna"

(earl/oil and loining/learning, respectively). These edits seem to indicate punners' awareness of rhoticity variation among English dialects (Labov, 1972).

Consonants. In Figure 2, we see several expected trends. D (d) ousting DH (ð), T (t) ousting TH (θ), and V (v) ousting W (w) are all as expected according to the "marked ousting unmarked" hypothesis of Zwicky and Zwicky (1986). Yet we also see S (s) ousting Z (z), which is an instance of the unmarked voiceless alveolar fricative ousting the voiced, as well as N (n) ousting NG (ŋ), an instance of the

unmarked coronal ousting the marked velar. These oustings, like the case of AH ousting other vowels, are likely conditioned by segment frequency. For N ousting NG, syllable structure may play a role as well, as NG is restricted to codas whereas N is not.

An interesting feature of our consonant data is the extreme violability of interdentals TH and DH, which are more likely to map to T and D respectively than to retain their identity in a paronomasic pun. In addition to the "fit of whiskey" example mentioned earlier, we have

"Disgusting wind knocked over my trash cans" (disgusting/this gusting). This phenomenon is known as "th-stopping" and is a common dialectal feature of many variants of English, from those of Philadelphia and New York to the Caribbean (Wells, 1982). This substitution supports the hypothesis that the substitution possibilities for a segment depend on the realizations of that segment which are commonly encountered in speech. Notably, T is significantly more likely to replace TH than is F (f), despite that the articulatory and acoustic similarity between TH and F is greater.

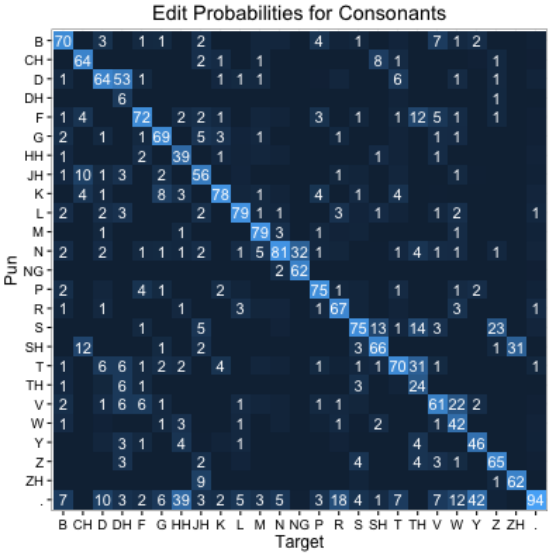

Figure 2: Edit probabilities for consonants based on the LM + PEM model.

5.2 Correlation with Human Ratings

Our phonetic edit model allows us to empirically verify an untested assumption with respect to phonological similarity from Fleischhacker (2002) "that degree of representation in the pun corpus correlates with pun goodness." In another paper, she repeats this assumption and adds the explanation "Truly funny puns are generally those in which the phonological relationship between pun and target is ... subtle but quickly recognizable" (Fleischhacker, 2005). Hempelmann writes that this assumption is "unlikely" to be true.

We conducted a survey of native English speakers where respondents were asked to rate 17 puns on a five point scale: hilarious, funny, okay, bad, terrible. The puns were selected from the test set to have a variety of phonemic edit distances. Respondents also had the option to indicate that they did not understand the pun, in which case their answer was ignored. A phonetic edit score was calculated for each pun-target pair by averaging the log-likelihood value from our model over the phonemes in the pun. The order of the questions was randomized for each respondent. Advertising for the survey was done using /r/SampleSize, a Reddit forum for recruiting survey participants. The 435 respondents gave us 7,135 ratings in total.

The relationship between phonetic edit cost and goodness was measured using ordinal regression, with clustered standard errors to account for the fact that responses from the same person are not independent. We use the RMS package in R (Harrell Jr., 2015). The regression coefficient indicates that decreased phonetic edit cost is indeed associated with higher perceived goodness of the pun with $p < 0.0001$. For visualization purposes we mapped the categorical goodness ratings onto a numeric scale from 1-5 to create an average goodness rating for each pun. In Figure 3, we depict the phonetic edit cost vs. the average goodness rating for each of the 17 puns. The line shown in that figure is an outlier resistant linear regression.

The biggest outlier is

> "They say a Freudian slip is when you say one thing but really mean your mother."

The pun is on "your mother" with "another" as the target. This pun has the highest phonetic edit cost in our sample but it makes up for it with more interesting semantics than average.

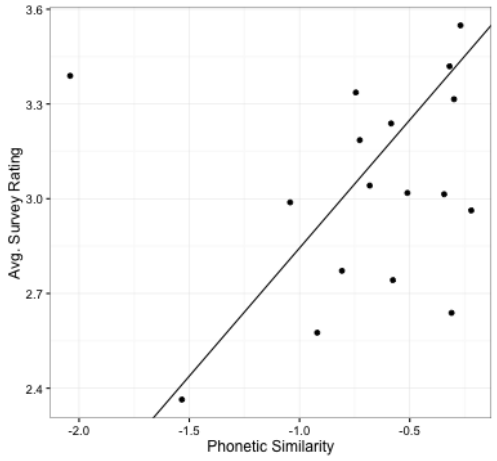

Figure 3: Relationship between the LM + PEM phonetic edit cost and goodness of the pun.

6 Conclusions and Future Work

The quality of our phonetic edit model is evident from its performance at the target recovery task, as well as the fact that it captures known linguistic phenomena such as vowel reduction and dialectal features. Furthermore, by collecting human ratings we are able to empirically verify the previously untested assumption that lower phonetic edit costs in puns correlate with pun goodness.

The strength of the model can be leveraged to improve the quality of pun generation and humor classification systems that have used weaker phonetic edit models (Binsted, 1996; Valitutti, 2011; Raz, 2012). Some pun generation systems are limited to exact homophones. In this work, we did not consider homographic puns. In principle, our algorithm can handle these by introducing an LM weight to control the balance of PEM/LM scores. Pun generation is much more complicated than target recovery as reflected in the complexity of proposed systems for humor generation. However, improved understanding of puns by way of progress in the target recovery task should also lead to corresponding improvements in the task of pun generation.

Our syllable extension to the PEM gave the best performance, but only by a small margin. Extending the edit model further is a fruitful area for future work but will likely require additional data.

In this work, we assume that the pun is given. Of interest for future work is joint recognition of the pun and its target. Preliminary experiments indicate that the unigram word probabilities are a somewhat strong feature for pun recognition but further work is needed. For contextually-integrated puns, identifying the pun is likely to be more difficult, and for some cases it would be useful to integrate image cues.

Acknowledgments

We thank Arjun Sondhi for his assistance in designing and analyzing the survey and Hope Boyarsky for her help assembling our pun corpus. We also would like to thank Nathan Loggins and the anonymous reviewers for their feedback on this work.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata*, pages 11–23. Springer.

Stephen R Anderson and David W Lightfoot. 2002. *The language organ: Linguistics as cognitive physiology*. Cambridge University Press.

Susan Bartlett, Grzegorz Kondrak, and Colin Cherry. 2009. On the syllabification of phonemes. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 308–316. Association for Computational Linguistics.

Kim Binsted. 1996. Machine humour: An implemented model of puns.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. " O'Reilly Media, Inc.".

Luigi Burzio. 2007. Phonology and phonetics of english stress and vowel reduction. *Language Sciences*, 29(2):154–176.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

John S Crosbie. 1977. *Crosbie's dictionary of puns*. New York: Harmony Books.

Katherine M. Crosswhite. 2004. Vowel reduction. In Bruce Hayes, Robert Kirchner, and Donca Steriade, editors, *Phonetically based phonology*, pages 191–231. Pearson Education.

Heidi Fleischhacker. 2002. Onset transfer in reduplication. In *LSA annual meeting. San Francisco: January*, pages 3–6.

Heidi Anne Fleischhacker. 2005. *Similarity in phonology: Evidence from reduplication and loan adaptation*. Ph.D. thesis.

Frank E Harrell Jr., 2015. *rms: Regression Modeling Strategies*. R package version 4.3-0.

Christian F Hempelmann. 2003. Paronomasic puns: Target recoverability towards automatic generation.

Takaaki Hori, Chiori Hori, Yasuhiro Minami, and Atsushi Nakamura. 2007. Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(4):1352–1365.

Sister Miriam Joseph. 2008. *Shakespeare's Use of the Arts of Language*. Paul Dry Books.

Stefan Daniel Keller. 2009. *The development of Shakespeare's rhetoric: a study of nine plays*, volume 136.

Grzegorz Kondrak. 2000. A new algorithm for the alignment of phonetic sequences. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 288–295. Association for Computational Linguistics.

William Labov, Sharon Ash, and Charles Boberg. 2006. Atlas of north american english: Phonology and phonetics. *Berlin: Mouton de Gruyter*.

William Labov. 1972. *Sociolinguistic patterns*. Number 4. University of Pennsylvania Press.

Tristan Miller and Iryna Gurevych. 2015. Automatic disambiguation of English puns.

M. Mohri, F. Pereira, and M. Riley. 2002. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Mehryar Mohri, Fernando Pereira, and Michael Riley. 2008. Speech recognition with weighted finite-state transducers. In *Springer Handbook of Speech Processing*, pages 559–584. Springer.

Jose Oncina and Marc Sebban. 2005. Learning unbiased stochastic edit distance in the form of a memoryless finite-state transducer. In *International Joint Conference on Machine Learning (2005). Workshop: Grammatical Inference Applications: Successes and Future Challenges*.

Yishay Raz. 2012. Automatic humor classification on Twitter. In *Proceedings of the NAACL Human Language Technologies: Student Research Workshop*, pages 66–70. Association for Computational Linguistics.

Eric Sven Ristad and Peter N Yianilos. 1998. Learning string-edit distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(5):522–532.

Graeme Ritchie. 2005. Computational mechanisms for pun generation. In *Proc. European Natural Language Generation Workshop*.

Włodzimierz Sobkowiak. 1991. *Metaphonology of English paronomasic puns*, volume 26. P. Lang.

Andreas Stolcke. 2000. Entropy-based pruning of backoff language models. *arXiv preprint cs/0006025*.

Keiko Tanaka. 1992. The pun in advertising: A pragmatic approach. *Lingua*, 87(1):91–102.

Alessandro Valitutti. 2011. How many jokes are really funny? towards a new approach to the evaluation. In *Human-Machine Interaction in Translation: Proceedings of the 8th International NLPCS Workshop*, volume 41, page 189.

Robert L Weide. 1998. The cmu pronouncing dictionary. *URL: http://www.speech.cs.cmu.edu/cgibin/cmudict*.

John C Wells. 1982. *Accents of English*, volume 1. Cambridge University Press.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. *EMNLP*.

Arnold Zwicky and Elizabeth Zwicky. 1986. Imperfect puns, markedness, and phonological similarity: With fronds like these, who needs anemones. *Folia Linguistica*, 20(3/4):493–503.