Testing Humor Theory Using Word and Sentence Embeddings

Stephen Skalicky
Victoria University of Wellington
Wellington, New Zealand
scskalicky@gmail.com

Salvatore Attardo
East Texas A&M University
Commerce, Texas, USA
salvatore.attardo@tamuc.edu

Abstract

A basic prediction of incongruity theory is that semantic scripts in verbal humor should be in a state of incongruity. We test this prediction using a dataset of 1,182 word/phrase pairs extracted from a set of imperfect puns. Incongruity was defined as the cosine distance between their word vector representations. We compare these pun distances against similarity metrics for the pun words against their synonyms, extracted from WordNet. Results indicate a significantly lower degree of similarity between pun words when compared to their synonyms. Our findings support the basic predictions of incongruity theory and provide computational researchers with a baseline metric to model humorous incongruity.

1 Introduction

Theories such as the General Theory of Verbal Humor describe the need for incongruity between elements of jokes, puns, and other humor forms, but this theory also stipulates there must a simultaneous degree of script overlap (Attardo and Raskin, 1991). As such, computational researchers should strive to model both opposition and overlap to better connect computational algorithms to humor theory (Hempelmann, 2008). In this paper, we do so by flipping the script on prior computational research which has used incongruity as a means to generate verbal humor (e.g., Ritchie, 2004; Mihalcea et al., 2010). Instead, we aim to determine the usefulness of recent advances in word vector representations to test some aspects of humor theory with the domain of puns.

1.1 Puns and Incongruity

Puns are defined as follows:

> A pun is a textual occurrence in which a sequence of sounds must be interpreted with a formal reference to a second sequence of sounds, which may, but need not, be identical to the first sequence, for the full meaning of the text to be accessed. The perlocutionary goal or effect of the pun is to generate the perception of mirth or of the intention to do so. (Attardo, 2020, p. 177–178)

In more accessible terms, two sequences of sounds evoke two meanings, associated with the first and second sequence, respectively. These are known as the *pun* and its *target*. For example, consider this very old pun: *Why did the cookie cry? Its mother was a* ***wafer/away for*** *so long*. In this pun, we have the string text *a wafer* (the pun), which sounds like the words *away for* (the target). Note in passing that outside of context, it does not matter which is the pun and which the target. Incongruity theory makes a clear prediction: the two senses (the pun and its target) should be in a relationship of incongruity. Incongruity is defined on the basis of semantic expectations. The (non–punning) sentence "I had a peanut butter and jelly sandwich" is congruous; the sentence "I had a peanut butter and jelly suitcase" is incongruous. In the case of puns, the incongruity has to reside, ex hypothesis, in the pun/target pair (since the rest of the text is identical). Let's consider again "why did the cookie cry?" Its mother was [a wafer]/[away for] so long" since the rest of text "why did the cookie cry?" Its mother was [. . .] so long" is identical in either reading, the incongruity can reside only in the pair "a wafer"/"away for."

1.2 Current Study

Our goal in this paper is to test this prediction, using the metric of cosine distance between word vector representations. Word vectors (or embeddings) capture semantic relationships as a function of distributional similarity. Words which appear in similar contexts have similar meanings, and their vector representations transform these relationships

Proceedings of the 1st Workshop on Computational Humor (CHum), pages 58–62
January 19, 2025.

into a numerical, multidimensional vector space (Mikolov et al., 2013). The angle between two vectors is commonly used as a measure of the difference between them and the cosine of the angle "has the nice property that it is 1.0 for identical vectors and 0.0 for orthogonal vectors" (Singhal et al., 2001, p. 3).

Distances among vectors have previously been applied to the problem of pun generation, where it was shown that vector distance could be used to model local and global similarity of pun words, improving pun generation performance (He et al., 2019). Here we also use distances between vectors, except we seek to measure the *similarity* between words in existing puns. Cosine vector similarity has been used to compare text similarity since the 1960s (Salton, 1963), and possibly earlier, in the context of information retrieval with keywords. More recently, cosine distance has been used as a measure of semantic acceptability/deviance (Vecchi et al., 2017), as well as metaphoricity and creativity (Winter and Strik-Lievers, 2023).

2 Method

We compare the cosine similarities of vector representation of key words used in puns. Our comparisons are made using both single–word vector spaces trained with word2vec (Mikolov et al., 2013) as well as sentence embeddings using the sentence transformers architecture (Reimers and Gurevych, 2019). We further compare the degree of these cosine distances between pun words against a baseline of their synonyms, collected from WordNet (Fellbaum, 1998)[1].

2.1 Materials

We utilized a corpus of 1182 pun–target word pairs analyzed in Hempelmann (2003), which were drawn from a subset of a larger corpus of puns discussed in Sobkowiak (1991). These 1182 pairs are from imperfect, heterophonic puns, meaning that the sound of the pun and target are not the same[2]. Using these puns, Hempelmann (2003) outlined a hierarchy of phonological, syntactic, and semantic constraints which differentiated between "good" and "bad" imperfect puns. While 959 of the pun–target pairs are in the form of a word–word relationship (e.g., *frozen* and *chosen*), 223 others represent a word–phrase relationship (e.g., *fundamental* and *from the mantel*).

2.2 Measuring Pun–Target Similarity

We calculated vector representations for each word/phrase in the pun–target pairs using two methods. Firstly, we used the `sentence-transforms` (SBERT) Python module to generate vector representations for pun words using the `all-MiniLM-L6-v2 model`. This is a lightweight model provided by Hugging Face, which was fine–tuned on over 1 billion sentence pairs with a 384–multidimensional vector space[3]. This sentence embedding model allowed us to calculate vectors for the 223 targets which spanned more than one word. However, recognizing the majority of the puns were pairs of single words, we also calculated a second set of similarities using the `word2vec-google-news-300` vector representations for single words using the `word2vec` algorithm, trained on 100 billion words from Google News corpus, creating a 300–dimensional space.[4]

Cosine distances between puns and targets for SBERT and word2vec representations were measured using the `cosine_similarity` function from the `scikit-learn` Python module. In addition to the 223 puns with targets more than one word long, an additional 153 puns contained a pun or target word not in the pre–trained word2vec space, meaning their pairwise similarity could not be calculated using word2vec. The average similarity between pun–target pairs was 0.270 (SD = 0.109) for the SBERT vectors, and 0.143 (SD = 0.203) for the word2vec vectors. For word2vec, some of the cosine distances were negative, so we also calculated the absolute values to better compare the degree of similarity (positive or negative) against the SBERT values. The results were closer, with the average absolute word2vec distances at 0.198 (SD = 0.150). We provide more examples across the full distribution in Table 1, and plot the density of the cosine distances in Figure 2.

We interpret these results as initial confirmation of incongruity theory. The maximum possible range of the absolute values is 0.0 (no relation) to 1.0 (completely related/completely unrelated). Finding that the median/average values for the puns is near the lower quartile of possible distances suggests that the occurrence of a semantically different

[1]We have made our code and data available on an OSF Repository

[2]Contrast these with homophonic puns, such as *I used to be a banker, until I lost* ***interest.***

[3]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[4]https://code.google.com/archive/p/word2vec/

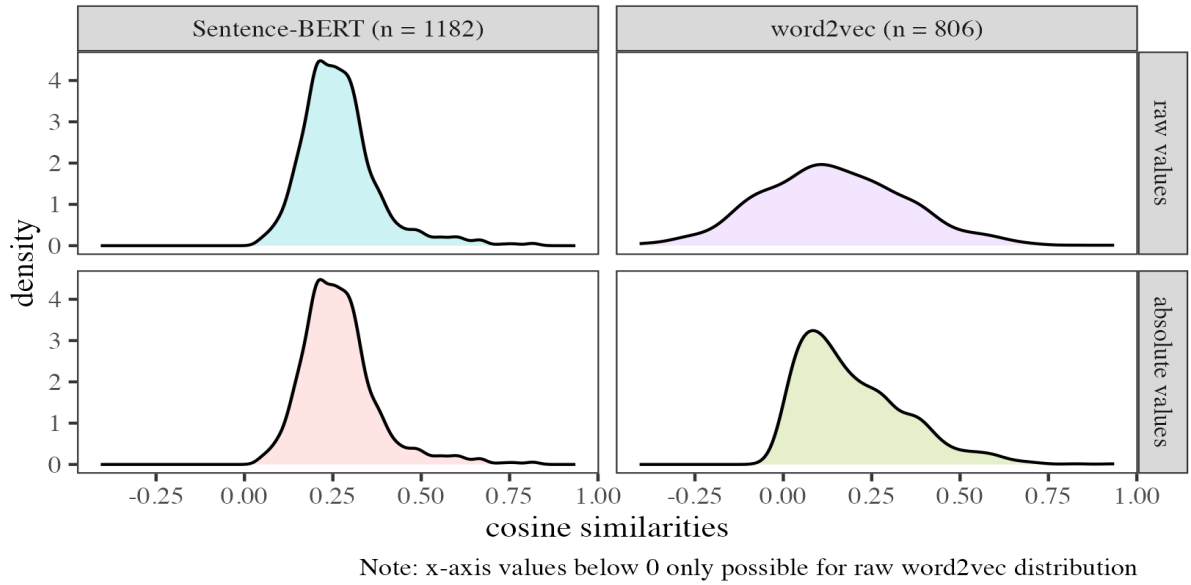

Figure 1: Distribution of cosine similarities between pun–target words using SBERT and word2vec.

Sentence–BERT			
pun	target	sim	range
cinnamon	see a man	0.046	*min*
kitty	pity	0.161	*−1SD*
prostituting	prosecuting	0.260	*median*
tinkle	twinkle	0.379	*+1SD*
invisible	visible	0.814	*max*
word2vec			
pun	target	sim	range
accost	cost	–0.407	*min*
salivation	salvation	–0.060	*−1SD*
pawn	upon	0.134	*median*
sheep	cheap	0.346	*+1SD*
thing	think	0.936	*max*

Table 1: Similarity values between pun and target words across distribution of similarities (minimum, median, and +/−1 standard deviation from mean), with example words.

sense will be unexpected and hence incongruous. As shown in Figure 2, most of the pun–target pairs live in this range, as indicated by the peak density values.

2.3 Establishing a Comparative Baseline

However, there is a difficulty: while in puns we can contrast the pun with its target, what should we compare that measure to? Although the resulting similarities for our pun–target pairs are relatively low on a scale from 0 to 1, a comparative baseline is needed in order to contextualise these results. We could compare a word to itself, but that only guarantees that we will get a score of approximately 0.99, or asymptotically tending to one.

As a solution, we queried WordNet, a large database of lexical meanings and connections for English words (Fellbaum, 1998). Each lexical entry in WordNet includes a list of *synsets*, which capture different semantic uses. For example, the entry *humour* has seven synsets: *temper*, *wit*, *liquid body substance*, and four different senses of *humour* (feeling humour, being humorous, sense of humour, and humorous mood). Each synset includes a list of words associated with that sense, called *lemmas*. The lemmas for *wit* are 'wit', 'humor', 'humour', 'witticism', and 'wittiness'.

We gathered the lemmas of all synsets for each pun–target word. Using the same SBERT model as we used for the puns, we then and calculated the average cosine distance between each pun word and its full set of synset lemmas. The assumption behind this approach is that punning pair words should have higher cosine similarity to their synonyms than to the pun–target words. This approach introduces a certain amount of noise — only about 2/3 of the pun words are in WordNet (again because some pun words are phrases or unorthodox spellings), and some WordNet entries are more detailed than others (meaning a greater number of synsets and lemmas for some words than others).

Specifically, of the 2,190 unique pun–target words and phrases in the pun data (some words were repeated across puns), only 1,520 words could

be found in WordNet (69.41%). Using this set, results of the similarity comparisons between a word and its synonym(s) is an average of 0.422 (median = 0.385, *SD* = 0.156, min = 0.039, max = 0.962). This is an increase of ~56% from the average SBERT similarities and an increase of ~112% from the average of the absolute values of the word2vec similarities. Paired–sample t–tests comparing SBERT pun similarities against SBERT synonym similarities indicated these differences were significant and with large effect sizes for both pun (mean difference = 0.158, 95%CI [0.144, 0.171], t = 23.325, df = 908, Cohen's d = 1.111, $p < .001$) and target (mean difference = 0.139, 95%CI [0.128, 0.151], t = 23.704, df = 843, Cohen'd d = 1.169, $p < .001$) words found in WordNet. This comparison provides further support for the incongruity theory. Namely, the pun average similarity of 0.27 (using sentence embeddings) is capturing some degree of incongruity, in that it is much lower than the similarity to related words (0.42).

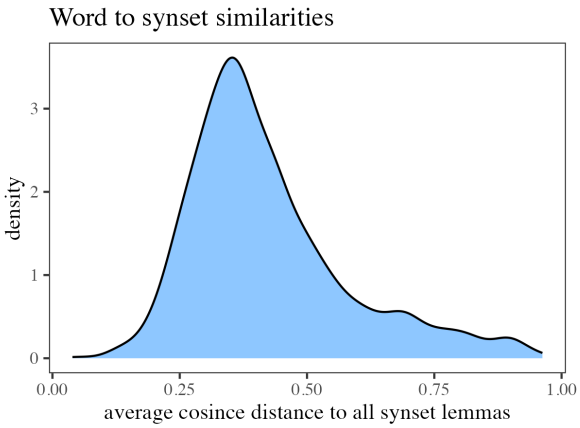

Figure 2: Distribution of cosine distance values between SBERT embeddings for pun words and their synset lemmas extracted from WordNet.

3 Discussion

The goal of our study was to test fundamental predictions of incongruity theory using computational measures of semantic distance. Using a dataset of pun–target words from 1182 imperfect, heterophonic puns, we found the cosine distances between word embeddings calculated using single–word vector spaces as well as sentence embeddings were on average significantly lower than those calculated from the pun words and their set of synonyms. In other words, pun–target words were less related to each other than they were to their own synonym entries in WordNet. As such, we find that the predictions based on the incongruity theory, that there should be a low degree of similarity between a pun and its target and that the similarity should be lower than that between a word and its synonyms, are borne out by our data.

It is noteworthy that single–word vectors from word2vec suggest less similarity between the pun words when compared to the SBERT embeddings. This likely reflects the difference in algorithms and size of training data. Nonetheless, both sets of embeddings measure the distributional probability of the target words within co–word contexts, so we have some confidence that these vectors are capturing the distributional properties of larger contexts within which the words appear. Our results thus seem to model the necessary balance for humorous incongruity — the simultaneous relationship of opposition and overlap. If the pun words were completely unrelated (e.g., cosine distances around 0), it would be unlikely that both words could be acceptable within the same sentence contexts, and thus the puns would simply not work.

We should of course point out the limitations of the study: first, we used a limited data set, collected by one scholar over 30 years ago. This may have introduced biases we are unaware of. Moreover, we used only the pun–target words from the puns — including full sentence contexts could provide more contextual information for baseline comparisons. Second, there are a growing number of different sentence embedding models, all of which will return different vector embeddings depending on their training data. More sophisticated models may be needed for future data sets and different types of humor, with the caveat that larger models comes at the expense of computational performance. Regardless, further replication of our results using other trained models and measures of similarity with puns and other datasets is necessary.

4 Conclusion

Our results provide empirical validation of theoretical assumptions related to predictions of incongruity theory. Specifically, we find incongruity between overlapping scripts of verbal humor as it occurs in puns using computational measures of semantic distance.

References

Salvatore Attardo. 2020. *The linguistics of humor: An introduction*. Oxford University Press.

Salvatore Attardo and Victor Raskin. 1991. Script theory revis(it)ed: Joke similarity and joke representation model. *Humor - International Journal of Humor Research*, 4(3-4):293–348.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. *arXiv preprint arXiv:1904.06828*.

Christian Hempelmann. 2008. Computational humor: Beyond the pun? In Victor Raskin, editor, *The primer of humor research*, pages 333–360. Mouton de Gruyter.

Christian F Hempelmann. 2003. *Paronomasic puns: Target recoverability towards automatic generation*. Ph.D. thesis, Purdue University.

Rada Mihalcea, Carlo Strapparava, and Stephen Pulman. 2010. Computational models for incongruity detection in humour. In *Computational linguistics and intelligent text processing*, pages 364–374. Springer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Preprint*, arXiv:1908.10084.

Graeme Ritchie. 2004. *The linguistic analysis of jokes*. Routledge, New York, NY.

Gerard Salton. 1963. Associative document retrieval techniques using bibliographic information. *J. ACM*, 10(4):440–457.

Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.

Włodzimierz Sobkowiak. 1991. *Metaphonology of English Paronomasic Puns*. P. Lang.

Eva M Vecchi, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2017. Spicy adjectives and nominal donkeys: Capturing semantic deviance using compositionality in distributional spaces. *Cognitive Science*, 41(1):102–136.

Bodo Winter and Francesca Strik-Lievers. 2023. Semantic distance predicts metaphoricity and creativity judgments in synesthetic metaphors. *Metaphor and the Social World*, 13(1):59–80.