Quantifying Humorous Lexical Incongruity

Chris Venour, Graeme Ritchie, Chris Mellish

University of Aberdeen

Abstract. Traditional discussions of humorous texts often postulate a notion of "incongruity" as being central, but there is no real formalisation of this notion. We are exploring one particular type of incongruity, in which a clash between the style or register of lexical items leads to humour. This paper describes our construction of a semantic space in which the distance between words reflects a difference in their style or tone. The model was constructed by computing profiles of words in terms of their frequencies within various corpora, using these features as a multidimensional space into which words can be plotted and experimenting with various distance metrics to see which measure best approximates differences in tone.

1 Introduction

The study of humour using computational techniques is still at a very early stage, and has mainly consisted of two kinds of project: the computer generation of very small humorous texts [1,2] and the use of text classification to separate humorous texts from non-humorous texts [3]. Little of this work has so far explored what many theories of humour claim is an essential ingredient of humour: incongruity[4,5][1]. On the other hand, non-computational humour research fails to construct clear and formal definitions of this concept. Our work seeks to bridge this gap, by creating and implementing a precise model of a simple kind of humorous incongruity.

The particular type of textual humour that we are focusing on, sometimes called register-based humour [4], is where the broader stylistic properties of words (in terms of style, social connotation, etc.) within a text are in conflict with each other. We intend to model this phenomenon by finding a semantic distance metric between lexical items, so that the intuition of 'words clashing' can be made precise. The semantic space we envision will provide an objective and quantifiable way of measuring a certain kind of humorous incongruity - a concept which has proven hard to measure or even define. The space we have developed is designed to automatically identify a particular class of jokes, and we plan to use it to generate original jokes of this type.

[1] Mihalcea and Strapparava [3] suggest that one of the features used by their classifier - antonymy - is a form of incongruity.

2 Incongruity theory

Incongruity theory is probably “the most widely accepted humour doctrine today (and) was born in the seventeenth century when Blaise Pascal wrote ‘Nothing produces laughter more than a surprising disproportion between that which one expects and that which one sees”’ [6]. The idea of incongruity has been variously defined in the literature - so much so that “it is not even obvious that all the writers on this subject have exactly the same concept in mind” [5] - but few commentaries offer more detail than the vague description left by Pascal.

Although some detailed work has been done describing some of the mechanisms of humorous incongruity - see the two-stage model [7] and the forced reinterpretation model described and extended by [5] - models such as these are still not specified enough to be implemented in a computer program. We hope to make some progress in this regard by creating a precise model of a certain kind of incongruity and implementing it to recognize a class of humorous text. The kind of humorous incongruity we formally model and then test in a computer program involves creating opposition along the dimensions of words.

3 Dimensions and Lexical Jokes

Near-synonyms, words that are close in meaning but not identical, reveal the kinds of subtle differences that can occur between words – nuances of style or semantics which make even words that share the same literal meaning slightly different from each other. For example the words ‘bad’ and ‘wicked’ are near-synonyms – both mean ‘morally objectionable’ – but differ in intensity. Similarly the words ‘think’ and ‘cogitate’ are almost synonymous but differ in terms of formality. These distinctions between near-synonyms – the ideas of ‘intensity’ and ‘formality’ in the examples above – are what we call dimensions. We believe that humorous incongruity can be created by forming opposition along these and other dimensions. To illustrate this idea, consider the following humorous text, taken from an episode of ‘The Simpsons’ (Sunday, Cruddy Sunday) in which Wally and Homer have been duped into buying fake Superbowl tickets:

> Wally: Oh, how could I fall for fake tickets? Gee, the fellas are gonna be crestfallen.

Instead of saying ‘disappointed’, Wally uses an outdated, highly literary and formal word, ‘crestfallen’. This choice of word smacks of a kind of effete intellectualism, especially in the highly macho context of professional sports, and the result is humorous. In choosing the word ‘crestfallen’, it is suggested that Wally mistakenly anticipates how ‘the fellas’ will react – with sadness rather than anger – but he has also chosen a word that is:

- noticeably more formal than the domain made salient by the scene (football)
- has an opposite score on some sort of ‘formality’ dimension than many of the other words in the passage (‘gee’, ‘fellas’, ‘gonna’)

This kind of incongruity, formed by creating opposition along one or more dimensions, is, we believe, the crux of a subclass of humour we call lexical jokes. Using the idea of dimensions, we aim to automatically distinguish lexical jokes from non-humorous text, and also to generate new lexical jokes. We believe that there is a significant subset of lexical jokes in which the relevant dimensions of opposition have something to do with formality, archaism, literariness, etc.; for brevity, we will allude to this cluster of features as "tone".

4 Creating a semantic space

As we do not know how the relevant dimensions are defined, how these dimensions are related, and how they combine to create incongruity, it is not feasible to simply extract ratings for lexical items from existing dictionaries. Instead, we have used the distribution of words within suitable corpora as a way of defining the tone of a word. For example, in Figure 1 the grey cells represent the frequencies of words (rows) in various corpora (columns): the darker the cell, the higher the frequency. The words 'person', 'make' and 'call' display similar frequency count patterns and so might be considered similar in tone. Whereas the pattern for 'personage' is quite different, indicating that its tone may be different.

More precisely, our proposed model works as follows:

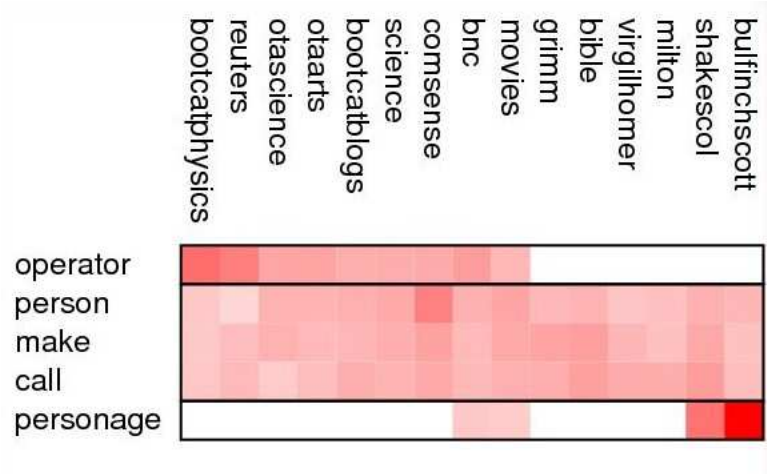
- select corpora which we judge to exhibit different styles or registers
- compute profiles of words in terms of their frequencies within the corpora
- use the corpora as dimensions, and the frequencies as values, to form a multidimensional space
- plot words from texts into the space
- try various outlier detection methods to see which one displays the outlier and clustering patterns we anticipate.

This model assumes that word choice is a significant determiner of tone. Syntax and metaphor, for example, may also play a very important role, but these are not considered here.

We looked for corpora which we think display differing degrees of formality/archaism/literariness. Besides using our intuition in this regard, we also felt that the age of a work is a strong determiner of how formal, etc. it sounds to modern ears, so we chose works that were written or translated in various time periods. Thus the following corpora were chosen for the first set of experiments: Virgil's The Aeneid (108,677 words), Jane Austen's novels (745,926), the King James version of the bible (852,313), Shakespeare's plays (996,280), Grimm's fairy tales (281,451), Samuel Taylor Coleridge's poetry (101,034), two novels by Henry Fielding (148,337), a collection of common sense statements (2,215,652), Reuter's news articles (1,614,077), a year's worth of New Scientist articles (366,393), a collection of movie reviews (1,298,728) and the written section of the British National Corpus (BNC World Edition) (80 million)[2].

[2] Frequency counts of a word in the BNC were taken from the CUVPlus dictionary, available at the Oxford Text Archive.

Fig. 1. Using frequency count patterns as 'tonal fingerprints'. Cells in the table represent the frequencies of words (rows) in various corpora (columns).



5 Automatically identifying an incongruous word

5.1 The development data

Twenty lexical jokes were used to develop the model. All contained exactly one word (shown in bold in the examples below) which we judged to be incongruous with the tone of the other words in the text[3].

1. *Operator, I would like to make a* ***personage*** *to person call please* (The Complete Cartoons of the New Yorker (CCNY), 1973, p.312).
2. *Sticks and stones may break my bones but* ***rhetoric*** *will never hurt me* (CCNY 1970, p.624).
3. *You cannot expect to wield supreme executive power just because some watery* ***tart*** *threw a sword at you (*Monty Python and the Holy Grail).
4. *Listen serving the customer is* ***merriment*** *enough for me* (The Simpsons, "Twenty-Two Short Films About Springfield").

Most of the jokes (15/20) are captions taken from cartoons appearing in the New Yorker magazine. Joke #3 however is taken from a scene in *Monty Python and the Holy Grail* and three of the twenty jokes are from different episodes of The Simpsons television show. Thus all the texts - except possibly one whose exact provenance is difficult to determine - are snippets of dialogue that were accompanied by images in their original contexts. Although the visual components enhance the humour of the texts, we believe the texts are self-contained and remain humorous on their own.

[3] A more formal test with volunteers other than the authors will be conducted in the future.

5.2 Computing scores

In the tests, stopwords were filtered from a lexical joke, frequencies of words were computed in the various corpora (and normalized per million words) and were treated as features or dimensions of a word. Words were thus regarded as vectors or points in a multi-dimensional space and the distances between them computed. We are interested in finding outliers in the space because if position in the space is in fact an estimate of tone, the word furthest away from the others is likely to be the word whose tone is incongruous.

Ranked lists of words based on their mutual distances (using different distance metrics described below), were therefore computed. If the word appearing at the top of a list matched the incongruous word according to the gold standard, a score of 2 was awarded. If the incongruous word appeared second in the list, a score of 1 was awarded. Any results other than that received a score of 0.

The baseline is the score that results if we were to randomly rank the words of a text. If a text has 9 content words, the expected score would be 2 * 1/9 (the probability of the incongruous word showing up in the first position of the list) plus 1 * 1/9 (the probability of it showing up second in the list), yielding a total expected score of 0.33 for this text. This computation was performed for each text and the sum of expected scores for the set of lexical jokes was computed to be 9.7 out of a maximum of 40.

5.3 Computing the most distant word in a text using various distance metrics

Different methods of computing distances between words were tried to determine which one was most successful in identifying the incongruous word in a text. Our first set of experiments, performed using the corpora listed above, employed three different distance metrics:

1. Euclidean distance: this distance metric, commonly used in Information Retrieval [8], computes the distance D between points $P = (p_1, p_2, \ldots p_n)$ and $Q = (q_1, q_2, \ldots q_n)$ in the following way:

$$D = \sqrt{\sum_{i=i}^{n} (p_i - q_i)^2}$$

   A word's Euclidean distance from each of the other words in a lexical joke was calculated and the distances added together. This sum was computed for each word and in this way the ranked list was produced. The word at the top of the list had the greatest total distance from the other words and was therefore considered the one most likely to be incongruous.
2. Mahalanobis distance: This distance metric, considered by [8] as "one of the two most commonly used distance measures in IR" (the other one being Euclidean distance according to these same authors), is defined as

$$D^2 = \sum_{r=1}^{p} \sum_{s=1}^{p} (x_r - \mu_r)\, v^{rs}\, (x_s - \mu_s)$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_p)$, μ is the population mean vector, $\mathbf{V}$ is the population covariance matrix and $v^{rs}$is the element in the rth row and sth column of the inverse of $\mathbf{V}$. For each word in a text, the Mahalanobis distance between it and the other words in the text is computed and the ranked list is produced.

3. Cosine distance: Another method of estimating the difference in tone between two words, regarded as vectors v and w in our vector space, is to compute the cosine of the angle θ between them:

$$cosine\,(\theta) = \frac{v \cdot w}{\|v\| \cdot \|w\|}$$

Cosine distance is commonly used in vector space modelling and information retrieval [9] and was used here to produce a ranked list of words in the manner described in 1. above.

5.4 Initial results

Table 1 shows the outcomes of testing on development examples using the set of corpora A (listed in Section 4) and various distance metrics. Predicting the incongruous word in a text using Euclidean distances received a low score of 2 out of a maximum of 40 and proved to be worse than the baseline score. Computing the most outlying word in a text with the Mahalanobis metric yielded a score of 11 which is only slightly better than random, while using cosine distances yielded the best result with a score of 24.

Table 1. Results from first set of testing

Test no.	Pre-processing	Distance metric	Corpora	Score (out of 40)
1	none	Euclidean	A	2
2	none	Mahalanobis	A	11
3	none	cosine	A	24

5.5 Experimenting with pre-processing

We experimented with two kinds of pre-processing which are familiar in information retrieval:

1. tf-idf: In an effort to weight words according to their informativeness, tf-idf [10] changes a word's frequency by multiplying it by the log of the following ratio: (the total number of documents)/(how many documents the word appears in). This transformation gives a higher weight to words that are rare in a collection of documents, and so are probably more representative of the

documents to which they belong. Our model computes frequency counts in corpora rather than documents, however, so the ratio we use to weight words is a variation of the one normally computed in information retrieval.

2. log entropy: When we compute the frequencies of words in the various corpora, the data is stored in a frequency count matrix $\mathbf{X}$ where the value of the cell in row i and column j is the normalized frequency count of word i in corpus j. Our second method of pre-processing, which has "been found to be very helpful in information retrieval" [11], involved computing the log entropy of the columns of matrix $\mathbf{X}$. This amounts to giving more weight to columns (i.e. corpora) that are better at distinguishing rows (i.e. words). Turney [11] describes how to perform this pre-processing.

Tf-idf transformations (Table 2) generated generally worse results. Log entropy pre-processing improved all the results however, the best result emerging once again from use of the cosine metric: its score improved from 24 to 32.

Table 2. Results from performing pre-processing

Test no.	Pre-processing	Distance metric	Corpora	Score (out of 40)
1	tf-idf	Euclidean	A	3
2	tf-idf	Mahalanobis	A	*4/36
3	tf-idf	cosine	A	14
4	log entropy	Euclidean	A	13
5	log entropy	Mahalanobis	A	23
6	log entropy	cosine	A	32

*Octave, the software we are using to compute the Mahalanobis distance, was for reasons unknown, unable to compute 2 of the test cases. Thus the score is out of 36.

5.6 Experimenting with different corpora

After achieving a good score predicting incongruous words using log entropy pre-processing and the cosine distance metric, we decided to not vary these methods and to experiment with the set of corpora used to compute frequency counts.

In experiment #1 corpus set B was built simply by adding four more corpora to corpus set A: archaic and formal sounding works by the authors Bulfinch, Homer, Keats and Milton. This increased the corpora size by ~600K words but resulted in the score dropping from 32 to 31 out of a maximum of 40.

In experiment #2 corpus set C was built by adding another four corpora to corpus B: Sir Walter Scott's "Ivanhoe", a collection of academic science essays written by university students, a corpus of informal blogs, and a corpus of documents about physics. As we see from Table 3, adding this data (~1.5 million words) improved the score from 31 to 35.

In corpus set C, archaic and formal sounding literature seemed to be over represented and so in experiment #3 a new corpus set D was created by combining Virgil's Aeneid with works by Homer into a single corpus as they are very similar in tone. Shakespeare and Coleridge's work were also merged for the same reason, as were the works by Bulfinch and Scott. In this way, fewer columns of the 'tonal fingerprint' consisted of corpora which are similar in tone. Also, works by Jane Austen and by John Keats were removed because they seemed to be relatively less extreme exemplars of formality than the others. These changes to the set of corpora resulted in a score of 37 out of a maximum of 40.

Table 3. Results from using different sets of corpora

Corpora set	B	C	D
Score	31	35	37

The decisions made in constructing corpora set D, indeed most of the decisions about which corpora to use as foils for estimating tone, are admittedly subjective and intuitive. This seems unavoidable, however, as we are trying to quantify obscure concepts in such an indirect manner. To the degree that our assumption that frequency counts in various corpora can be an estimate of a word's tone, the kind of experimentation and guesswork involved in constructing our semantic space seems valid.

Thus using corpus set D, log entropy pre-processing and cosine distance as our distance metric, produced excellent results: 37 out of a possible 40 on the development set, according to our scoring, in identifying the incongruous word in the set of lexical jokes. We found that we were even able to raise that score from 37 to 39/40 (97.5%) by not eliminating stopwords from a lexical joke i.e. by plotting them, along with content words, into the space. Incongruous words in lexical jokes tend not to be commonplace and so including more examples of words with 'ordinary' or indistinct tone renders incongruous words more visible and probably accounts for the small rise in the score.

6 Automatically distinguishing between lexical jokes and regular text

The next step is to determine whether the space can be used to detect lexical jokes within a collection of texts. One way of automating this classification would be to find the most outlying word and to look at how far away it is from the other words in the text. If the distance were to be above a threshold, the program would predict that the text is a lexical joke.

This approach was tested on a set of texts consisting of the development set of lexical jokes together with a sample of 'regular' i.e. non lexical joke texts: newspaper texts randomly[4] selected from the June 5 2009 issue of the Globe and Mail, a Canadian national newspaper. Complete sentences from the newspaper

[4] Newspaper sentences containing proper names were rejected in the selection process because names appear haphazardly, making estimation of their tone difficult.

were initially much longer than the lexical joke sentences - the average number of words in the lexical jokes set is 16.1 - so newspaper sentences were truncated after the 17th word.

For each text, the most outlying word was determined using the cosine method described above (with log entropy pre-processing) and the average cosine (λ) it forms with the other words in the text was computed. Precision is highest when the threshold cosine value is arbitrarily set at 0.425 - i.e. when we say that λ needs to be less than or equal to 0.425 in order for the text to be considered a lexical joke. From Table 4 we see that 77.8% precision (in detecting jokes from within the set of all the texts processed) and 70% recall result using this threshold. (When pathological cases[5] are excluded from the evaluation, the program achieves 10/13 (76.9%) precision and 10/16 (62.5%) recall using this threshold).

Table 4. precision and recall when computing averages

threshold value	precision	recall	F score
<=0.5	19/26 (73.1%)	19/20 (95%)	82.6
<=0.425	14/18 (77.8%)	14/20 (70%)	73.7

The semantic space was developed to maximise its score when identifying the incongruous word in a lexical joke, but it has limited success in estimating how incongruous a word is. We believe that differences in tone in lexical jokes are much larger than those in regular text but the semantic space achieves, at best, only 77.8% precision in reflecting the size of these discrepancies.

One reason for this might be that the set of corpora is simply not large enough. When the threshold is set at .425, the three newspaper texts (not containing a pathological word) mistakenly classified as lexical jokes are:

- *the tide of job losses washing across north america is showing signs of* ***ebbing****, feeding hope that...*
- *yet investors and economists are looking past the grim* ***tallies*** *and focusing on subtle details that suggest...*
- *both runs were completely sold out and he was so* ***mobbed*** *at the stage door that he...*

The most outlying words in these texts (shown in bold) appear only rarely in the set of corpora: the word 'ebbing' appeared in only three corpora, 'tallies' in two and 'mobbed' in only one corpus. None of the other words in the newspaper texts appear in so few corpora and perhaps these words are considered significantly incongruous, not because they are truly esoteric (and clash with more prosaic counterparts) but because the corpus data is simply too sparse.

[5] Pathological texts contain words which do not appear in any of the corpora. These words were 'moola', 'tuckered', 'flummery', 'eutrophication' and 'contorts'.

The problem may be more deeply rooted however. New sentences which no one has ever seen before are constructed every day because writing is creative: when it is interesting and not clichéd it often brings together disparate concepts and words which may never have appeared together before. Perhaps the model is able to identify relatively incongruous words with precision but is less able to gauge how incongruous they are because distinguishing between innovative word choice and incongruous word choice is currently beyond its reach.

7 Future work

Results look promising but future work will need to determine how the method performs on unseen lexical joke data. In early experiments, Principal Components Analysis (PCA) was performed on the frequency count data in an attempt to reduce the feature space into a space with fewer (and orthogonal) dimensions but initial results were disappointing. One reason for this might be that the corpora are too sparse to allow for much redundancy in the features, but further investigations into using PCA and other techniques for reducing the dimensionality of vector spaces (such as Latent Semantic Analysis) will be performed. Finally, experiments into using the vector space to generate original lexical jokes will be conducted.

References

1. Stock, O., Strapparava, C.: HAHAcronym: Humorous agents for humorous acronyms. Humor-International Journal of Humor Research **16**(3) (2003) 297–314
2. Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D., Black, R.: The construction of a pun generator for language skills development. Applied Artificial Intelligence **22**(9) (2008) 841–869
3. Mihalcea, R., Strapparava, C.: Learning to laugh (automatically): Computational models for humor recognition. Computational Intelligence **22**(2) (2006) 126–142
4. Attardo, S.: Linguistic theories of humor. Walter de Gruyter, Berlin (1994)
5. Ritchie, G.: The linguistic analysis of jokes. Routledge, London/New York (2004)
6. Friend, T.: What's so funny? The New Yorker **November 11** (2002) 78–93
7. Suls, J.: A two-stage model for the appreciation of jokes and cartoons: an information-processing analysis. In: The Psychology of Humor: Theoretical Perspectives and Empirical Issues (Goldstein and McGhee). Academic Press, New York (1972) 81–100
8. Li, X., King, I.: Gaussian mixture distance for information retrieval. In: Proceedings of the International Conference on Neural Networks. (1999) 2544–2549
9. Salton, G., McGill, M.: Introduction to modern information retrieval. McGraw-Hill, New York, NY (1986)
10. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management **24**(5) (1988) 513–523
11. Turney, P.: Similarity of semantic relations. Computational Linguistics **32**(3) (2006) 379–416