A Computational Model of Lexical Incongruity in Humorous Text

Chris Venour

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of the
University of Aberdeen.

Department of Computing Science

June 2013

Declaration

I hereby declare that I alone composed this thesis and that its content is the result of my own research. No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Signed:

Date: June 2013

Abstract

Many theories of humour claim that incongruity is an essential ingredient of humour. However this idea is poorly understood and little work has been done in computational humour to quantify it. For example classifiers which attempt to distinguish jokes from regular texts tend to look for secondary features of humorous texts rather than for incongruity. Similarly most joke generators attempt to recreate structural patterns found in example jokes but do not deliberately endeavour to create incongruity.

As in previous research, this thesis develops classifiers and a joke generator which attempt to automatically recognize and generate a type of humour. However the systems described here differ from previous programs because they implement a model of a certain type of humorous incongruity.

We focus on a type of register humour we call lexical register jokes in which the tones of individual words are in conflict with each other. Our goal is to create a semantic space that reflects the kind of tone at play in lexical register jokes so that words that are far apart in the space are not simply different but exhibit the kinds of incongruities seen in lexical jokes. This thesis attempts to develop such a space and various classifiers are implemented to use it to distinguish lexical register jokes from regular texts. The best of these classifiers achieved high levels of accuracy when distinguishing between a test set of lexical register jokes and 4 different kinds of regular text.

A joke generator which makes use of the semantic space to create original lexical register jokes is also implemented and described in this thesis. In a test of the generator, texts that were generated by the system were evaluated by volunteers who considered them not as humorous as human-made lexical register jokes but significantly more humorous than a set of control (i.e.non-joke) texts. This was an encouraging result which suggests that the vector space is somewhat successful in discovering lexical differences in tone and in modelling lexical register jokes.

Acknowledgements

I would like to thank my supervisors Graeme Ritchie and Chris Mellish for their constant support, expert advice and great kindness. I cannot imagine better supervisors and feel very fortunate to have been their student.

Thank you to my fellow students Matthew Dennis and Roman Kutlak for generously helping me get the Mechanical Turk system to work. Thanks also to Robert Mankoff for sending me hundreds of New Yorker captions to be used for testing.

Thank you to all my family for their love and support. Special thanks to my brother David who taught me how to write essays a long time ago now. My education probably would have ended prematurely without his incredibly generous and patient help.

Finally, I would like to thank my dear parents Ralph and Jean Venour. Thank you for always being so encouraging and supportive. This dissertation is dedicated to you, with all my love and gratitude.

Contents

Chapter 1

Introduction

1.1 Motivation

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 small subclasses of humour (Stock and Strapparava, 2003; Manurung et al., 2008), and the use of text classification to separate humorous texts from non-humorous texts (Mihalcea and Strapparava, 2006). Little of this work has so far explored what many theories of humour claim is an essential ingredient of humour: incongruity (Attardo, 1994; Ritchie, 2004). On the other hand, non-computational humour research fails to construct clear and formal definitions of this concept (Ritchie, 1999, 2004). This thesis attempts to address some of these problems by creating and implementing a precise model of a simple kind of humorous incongruity.

The type of textual humour we focus on, sometimes called "register-based" humour (Attardo, 1994), is where the tones of words are in conflict with each other. We 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 build is intended to provide an objective and quantifiable way of measuring a certain kind of humorous incongruity. The space we have developed is designed to automatically identify a particular class of jokes, and it is also used to generate original jokes of this type.

1.2 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"' (Friend, 2002). 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" (Ritchie, 2004) - 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 (Suls, 1977) and the forced reinterpretation model described by Ritchie (2004) - models such as these are still not specified enough to be implemented in a computer program. We make some progress in this regard by creating a precise model of a certain kind of incongruity and implementing it to recognize and generate a class of humorous text. The kind of humorous incongruity we formally model and then test in both a classifier and generator involves creating opposition along the dimensions of words.

1.3 Dimensions and lexical register jokes

Near-synonyms are words that are close in meaning but not identical, and they reveal the kinds of subtle differences that can occur between words - nuances of style or semantics which make even words that share the same 'dictionary' 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. Our thesis is 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')

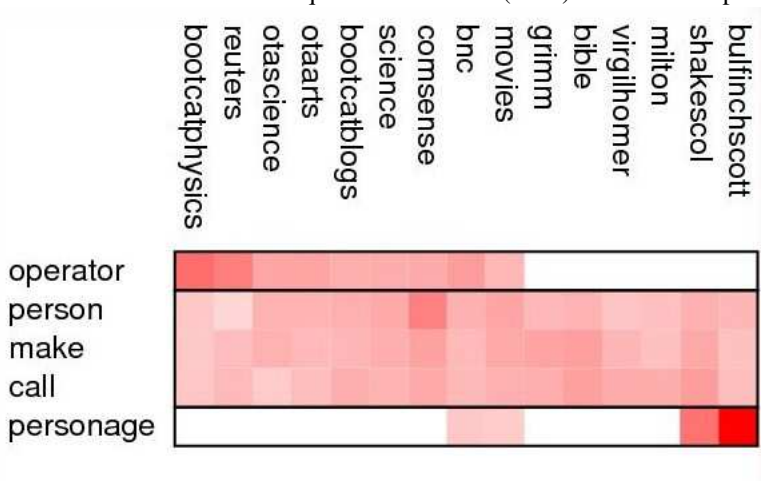
This kind of incongruity, formed by creating opposition along one or more dimensions, appears to be the crux of a subclass of humour we call *lexical register jokes*. Using the idea of dimensions, we aim to automatically distinguish lexical register jokes from non-humorous text, and also to generate new lexical register jokes.

1.4 A semantic space for recognizing and generating lexical register jokes

We believe that there is a significant subset of lexical register 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. As we do not know exactly how these dimensions are defined, how they are related, and how they combine to create incongruity, it is not feasible to simply extract ratings for lexical items from existing dictionaries. Instead, the distribution of words within suitable corpora are used as a way of defining the tone of a word. For example, in Figure 1.1 the 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 methodology for constructing the vector space works as follows:

Figure 1.1: Using frequency count patterns as 'tonal fingerprints'. Cells in the table represent the frequencies of words (rows) in various corpora (columns).



- select corpora which we judge to exhibit different styles or registers
- create profiles of words by computing 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 distance metrics to see which one displays the anticipated clustering patterns.

Five different classifiers are implemented in this thesis and four of them make use of this vector space to automatically distinguish lexical register jokes from other kinds of text. (Classifier #2, which is introduced in Chapter 4, estimates differences in tone in a significantly different way than the others. It analyses co-occurrence statistics of words to estimate their difference in tone).

A generator was also built and makes use of the vector space to generate lexical register jokes. This system attempts to modify seed texts (i.e. regular texts that are not lexical register jokes) in such a way that at least one pair of words in a text are significantly far apart from each other in the space. Volunteers were asked to evaluate the output of the generator and scores show that the generated texts were considered not as humorous as human-made lexical register jokes but were regarded as significantly more humorous than non-joke texts.

1.5 Thesis outline

The thesis is organized in the following way:

- Chapter 2 describes previous research that is relevant to this thesis. For example previous work on register humour, near-synonyms, incongruity, joke generation, and classification is discussed.
- Chapter 3 describes our attempts at building a semantic space which reflects the tone of words and introduces a vector space classifier (classifier #1) which aims to distinguish lexical register jokes from regular text.

- Chapter 4 discusses the extent to which corpus sparsity might be responsible for classifier errors and explores what might be done to correct these errors. A second classifier (classifier #2) which uses the web as a giant corpus to estimate the co-occurrence of words is also introduced.
- Chapter 5 provides a more precise model of lexical register jokes and describes three new vector space classifiers (classifiers #3 - #5) which align better with this new model.
- Chapter 6 evaluates the performance of the five classifiers that have been developed up to this point, in a final test on unseen data.
- Chapter 7 describes how the best performing classifier (classifier #4) was modified into a system that generates lexical register jokes.
- Chapter 8 discusses a final test of the generator and assesses its performance.
- Chapter 9 discusses how the generator might be improved and extended and explores whether incongruity of tone alone is a sufficient condition for humour.
- Chapter 10 offers some final conclusions about the work.

Chapter 2

Related research

2.1 Humour theory

2.1.1 Incongruity

Many theories of humour (for example Koestler (1964); Keith-Spiegel (1972); Suls (1983)) claim that incongruity is an essential ingredient of humour. But as Ritchie (2004) argues in a review of the literature on this topic, these theories fail to construct clear and formal definitions of the concept. One of the few attempts to make the idea of incongruity tangible is Nerhardt (1970). In this paper, Nerhardt describes a set of experiments in which a series of increasingly heavier weights were placed in the hand of a subject, whose eyes were closed, in order to create an expectation about how heavy the next weight would be. (The experiments only investigate this direction of change i.e. Nerhardt does not test whether increasingly lighter weights, ending with a final heavy weight, have a humorous effect).

The aim of the experiment was to determine whether people would laugh when they were given a weight that "deviated ... from what they expected" and the author hypothesized that "the frequency of laughter is a function of the discrepancy between expectations and actual states, a higher frequency being connected with greater discrepancy up to a certain point" (Nerhardt, 1970). Results of the experiments supported his hypothesis and it was determined that "(t)he greater the divergence of a stimulus from expectation in one or many dimensions, the funnier the stimulus" (Nerhardt, 1970). He concludes that "variation in incongruity alone ... is capable of eliciting the experience of humor".

This experiment raises an important issue. It suggests that perhaps all that is required for humour is to create an expectation which is then defeated. Yet not all surprising things are humorous. Our research explores this subject to some extent by asking the following question: is incongruity of tone, which is a kind of surprise, sufficient for generating a type of humour or does something else have to exist in a text? Veale (2004) addresses this issue when he suggests that incongruity may not be a "root cause" of humour but is merely 'epiphenomenal' i.e. a secondary phenomenon that often occurs in jokes but is not primarily responsible for their humour. Veale argues that to explain why a text is funny, "a humour theory must not look to incongruities but provide a social explanation for why we enjoy insulting others". Chapter 9 explores this topic in more detail.

2.1.2 Bisociation

Koestler (1964) uses the term bisociation rather than incongruity and defines bisociation as "the processing of a situation or idea ... in two self-consistent but habitually incompatible frames of

reference". Ritchie (2004) points out, however, that the theory is unclear about the meaning of the concepts it uses such as 'frames' , "perceive in" and "habitually incompatible". Ritchie (2004) asks, for example, "how does viewing something as frame-conflict tell us anything we would not glean from viewing it as some other form of logical clash (such as the overriding of default inference) or even just informally saying that there has been a clash between two perspectives?". The frame data structure, for example, "can be seen to be so general that it makes very little claim about how the knowledge is structured. The answer to 'what is a frame?' is virtually anything" (Ritchie, 2004).

Furthermore, the theory is unclear about how "humorous bisociations differ from other alignments or clashes between frames that might occur in, for example, analogies, misunderstandings, or poetic figures of speech (metaphor, simile, etc.)".

Lexical register jokes can be modelled in terms of bisociation theory - two incompatible frames of reference are evoked by a text with formal and informal tone - but looking at the jokes in this way only confuses our model (which is fully described in Chapter 5) by introducing the vaguely defined concept of frames.

2.1.3 Script opposition

Incongruity is described as "script opposition" in the "Semantic Script-based Theory of Humour" (SSTH) (Raskin, 1985) and also in the latter theory's reincarnation as the "General Theory of Verbal Humour" (GTVH) (Attardo and Raskin, 1991).

The SSTH argues that a text is a joke if the text is compatible with two different scripts but the scripts are somehow 'opposite'. Later descriptions of script opposition in the GTVH (see for example Attardo et al. (2002)) argue that a script "has an internal hierarchy so that some parts are more salient or foregrounded than others" (Ritchie, 2009b) and that two script are opposed (and therefore generate humour) if some of their foregrounded parts differ.

Lexical register jokes could perhaps be described in terms of script opposition but doing so confers no benefit to our model or implementations. Like the concept of 'frames' in bisociation theory, the notion of a script in the SSTH and the GTVH is so vague that it "could be any type of knowledge structure whatsoever" (Ritchie, 2004) and making use of this undeveloped concept would only obscure our model of these kinds of jokes.

2.1.4 Conceptual blending

The theory of "conceptual blending", first described by Fauconnier and Turner (1994), was originally proposed to model 'general' cognitive processes such as "categorization, the making of hypotheses, inference, analogy, metaphor and narrative". More recently, however, the theory has also been used to model humorous texts (see for example Kyratzis (2003); Lundmark (2003)).

Traditionally two-space models of expressions such as metaphors have been proposed in which a concept (called the 'target') is described in terms of another concept (the 'source') (see for example Lakoff and Johnson (2003)). For example expressions such as "he shot down all my arguments" and "your claims are indefensible" reveal a common conceptual metaphor that "argument is war". Here the concept of an argument (the target) is understood and "partially structured by the concept of war" (the source) (Lakoff and Johnson, 2003).

The theory of conceptual blending argues that two-space models cannot explain certain phenomena, however. For example two-space models fail to describe metaphors which have features that are not present in either the source or target domains. The metaphor "his surgeon is a butcher", for example, refers to an incompetent surgeon "but the concept of incompetence is not present" in either the surgeon or butcher domains (Kyratzis, 2003).

The theory of conceptual blending therefore introduces the idea of multiple "mental spaces" and most importantly a "blended space", which combines elements from the target and the source domains, which are re-labelled as "input spaces". Although the blended space "is created with elements from the input spaces", it is possible for it to have "its own emergent structure" (Kyratzis, 2003). Thus, in our example, the cruder instruments and practises of the butcher's trade are mapped into a new space, as are the tools and practises of the surgeon's craft, which require more precision and care. From this new blended space the notion of incompetence emerges.

Some theorists argue that the idea of conceptual blending "appears to be perfectly suited for the analysis of humorous texts" (Kyratzis, 2003). They propose that a blended space yields a metaphor if the boundary or line separating two disparate concepts is somehow "obliterated and the two items fuse to form a single entity" whereas a joke is created if the "boundary or line separating" two joined items is emphasized (Kyratzis, 2003).

Like bisociation and script opposition, the idea of conceptual blending, although interesting, is too vague and undeveloped to be of theoretical or practical use in this thesis. For example the concept of "mental spaces", like the idea of 'frames' in bisociation theory and 'scripts' in the SSTH and GTVH, is too abstract and undefined to be used in a computational model and important details about how the 'boundary' separating two items might be 'obliterated', to create a metaphor, or 'emphasized', to create a joke, are not provided by the theory.

2.1.5 Incongruity resolution theories

Incongruity resolution (IR) (Suls, 1977; Katz, 1993; Martin, 2007) is a popular theory of humour which states, that humour depends not only on the presence of incongruity but that that the incongruity has to be somehow "resolved or made sense of" (Oring, 2003).

Ritchie (2004) argues however that there is no "single, definitive statement" of the theory and that a number of different versions have been proposed. Ritchie explains that these versions of IR theory differ in terms of the following features:

1. scope: some researchers argue that all forms of humour involve incongruity resolution whereas others "merely claim that IR gives a good description of some subclass of humour".

2. sequentiality: a "widespread informal idea of IR" is that jokes create "an ordered sequence of events, with the audience first perceiving the incongruity, then grasping the resolution". However "a non-process-oriented variant of IR is not only logically possible" but has been proposed for example by Oring (2003).

3. location of incongruity: the various incongruity resolution theories propose that a clash of concepts occurs within humour but opinions vary about where this conflict takes place .

4. extent of resolution: opinions differ about whether the resolution wholly or only partially removes the incongruity.

The GTVH, for instance, can be regarded as an incongruity resolution theory which is "a sequential model, the scope of which is all verbal humour" (Ritchie, 2004). One version of GTVH (there appears to be many) claims that script opposition creates incongruity and that something called "Logical Mechanism" (LM), "which is rather under-defined", constitutes the resolution. LM is described as an optional feature of jokes and so this version of GTVH would be classified as an IR theory which argues for "partial or full resolution".

The theory of humour in Oring (2003) states that humour is the result of "appropriate incongruity" and it too can be regarded as an IR theory, with the following parameters:

1. scope: all of humour
2. sequentiality: non-sequential ("there is no need for the incongruity to be perceived prior to the resolution" Ritchie (2009b)).
3. location of incongruity: "as this is a non-sequential framework, it is not meaningful to ask whether the incongruity is related within the processing" (Ritchie, 2009b).
4. extent of resolution: partial ("Oring's notion of 'appropriate' is a form of partial resolution" Ritchie (2009b)).

The type of humour modelled in this thesis can be regarded as an example of sequential incongruity resolution: a reader encounters at least one incongruity of tone (which can take place anywhere in the text) and, it appears that if this incongruity is resolved in a certain way (i.e. a particular kind of explanation for it is found), the result is humorous. See Chapter 9 for details.

2.2 Computational approaches

From the literature (Lessard and Levison, 1992, 1993, 2005; Manurung et al., 2008) we see that there is methodological precedence for performing small scale studies of subclasses of humour. Developing detailed descriptions of particular types of humour may be a better approach than immediately attempting to create a general theory of all humour because the latter might easily "fail either in its generality (i.e. not cover all the possibilities) or in its accuracy (i.e. cover some data incorrectly)" (Ritchie, 2004). Some detailed models of particular subclasses of humour have been constructed and a handful of joke classifiers and generators have been created. This thesis implements both kinds - joke classifiers and a generator - and they make use of a vector space to estimate the tone of words.

2.2.1 Vector space models

The generator and most of the classifiers implemented in this thesis make use of a vector space in which the position of a word, which appears as a vector in the space, is an estimate of its tone. Turney et al. (2010) argue that vector space models (VSMs) are "arguably the most successful approach to semantics". Their paper, which is a survey of how these models have been used to solve various problems in semantics, identifies three classes of VSMs:

1. VSMs which determine the similarity of documents. Example applications perform tasks such as document retrieval, clustering, classification, essay grading, question answering, and call routing.

2. VSMs which determine the similarity of words. Example applications perform tasks such as word similarity, clustering, classification, automatic thesaurus generation, word sense disambiguation, context sensitive spelling correction and semantic role labelling.

3. VSMs which determine the similarity of relations. Example applications measure pattern similarity, and perform tasks such as automatic thesaurus generation and analogical mapping (e.g. "mason is to stone as carpenter is to x" where x = wood).

The first class of VSMs listed above builds a *term-document* matrix. If a document collection contains *n* documents and *m* unique terms, the term-document matrix X is built which has *m* rows - "one row for each unique term in the vocabulary" - and *n* columns - "one column for each document" (Turney et al., 2010). The simplest kind of matrix X contains frequencies - each row consists of how many times a word appears in each document. Columns of the matrix are therefore document vectors and an assumption made by the model is that vectors that cluster together in the space are likely to be about similar topics. The system described in Salton (1971) "was arguably the first practical, useful algorithm for extracting" this kind of semantic information using a term-document matrix.

Turney et al. explain that Deerwester et al. (1990) "were inspired by the term-document matrix of Salton et al." but focused instead on the row vectors of a frequency matrix rather than the column vectors in order to determine the similarity of words. A document "is not necessarily the optimal length of text for measuring word similarity", however, and so VSMs that belong to the 2nd class listed above, build a *word-context* matrix in which "the context is given words, phrases, sentences, paragraphs, chapters, documents, or more exotic possibilities such as sequences of characters or patterns" (Turney et al., 2010). The VSM developed in this thesis belongs to this 2nd class of VSM and the contexts it uses to estimate the similarity of word tone are corpora which vary in size from 150,000 to 1.7 million words.

The third class of VSM builds a *pair-pattern* matrix in which "row vectors correspond to pairs of words, such as *mason:stone* and *carpenter:wood*, and column vectors correspond to the patterns in which the pairs co-occur, such as "X cuts Y", and "X works with Y". This kind of matrix was first introduced by Lin and Pantel "for the purpose of measuring the similarity of patterns; that is, the similarity of column vectors". As Turney et al. explain "patterns which co-occur with similar pairs tend to have similar meanings" and so "pattern similarity can be used to infer that one sentence is a paraphrase of another".

All the VSMs outlined above make what Turney et al. call the "distributional hypothesis": "words that occur in similar contexts tend to have similar meanings". The vector space model developed in this thesis makes a similar assumption - that words with similar tone tend to appear in the same contexts. Other assumptions underlying our model are described in Section 3.3.1 of Chapter 3.

2.2.2 Previous generators

Nearly all previously built joke generators create puns that play with homophones i.e. words that sound or are spelt similarly (Lessard and Levison, 1992, 1993, 2005; Binsted, 1996; Manurung et al., 2008). These systems tend to rely on the vagaries of chance – the accident of homophony - to create a humorous clash of ideas. Results suggest, however, that the nature of the opposition

or clash of ideas that occur in puns and in many types of humour is important and needs to be considered. For example if just any pair of homophones is indiscriminately brought together in a pun, as they are in previous implementations, the resulting clash of ideas is more likely to be odd rather than humorous because homophones, although different, are not necessarily incongruous in a way that leads to humour. What that way is - i.e. the difference between "different ideas" and "humorously incongruous ideas" - is still a mystery and one that computational humour hopes to solve one day.

A system that does not generate puns - the HAHAcronym system (Stock and Strapparava, 2003) - takes as input an abbreviation (such as MIT) and its meaning (e.g. "Massachusetts Institute of Technology") and attempts to output an alternative phrase which is amusing (e.g. "Mythical Institute of Theology"). The program substitutes a subset of the words in the original phrase with words that have roughly the same rhythm but which belong to an 'opposing' domain. A list of opposing domains (such as religion vs. technology) was hand built and a lexicon of words annotated (again by hand) with domain labels was used by the program to make the word substitutions.

The generator implemented in this thesis does not assemble words that have been pre-selected as incongruous (as in the HAHAcronym system), nor does it attempt to build contexts for homographs that often have widely divergent meanings (as in the pun generators). Instead, its mandate is to automatically estimate the tone of words and create a clash of tone along the formality dimension in each of the texts it generates. Chapter 9 speculates about whether this is sufficient to generate humour. This thesis is also unique in that a model of a type of humorous incongruity is implemented and used to automatically recognize and generate a certain type of humour whereas previously built systems are not based on models of incongruity.

2.2.3 Previous classifiers

Mihalcea and Strapparava (2005) argue that previous work in computational humor "has focused mainly on the task of humor generation" and "few attempts have been made to develop systems for computational humor recognition". In this paper they claim that they are the first to use machine learning techniques to recognize humorous texts. Text categorization is "one of the testbeds of choice for machine learning" but presumably automatic classifiers had never been tested on humorous texts until 2005.

The goal of their initial experiments was to build an automatic classifier which can distinguish between one-liners and "structurally and stylistically similar" non-humorous texts of the same length: sentences from the BNC or Reuters. Two kinds of automatic classifiers frequently used for text classification were tested in their experiments: a Naive Bayes classifier and a Support Vector machine. The best results achieved were accuracies of 77.84% in distinguishing one-liners from BNC texts (using SVM) and 96.89% in separating one-liners from Reuters texts (using Naive Bayes).

Mihalcea and Strapparava's research is similar to the work described in this thesis because a semantic space is used to distinguish between humorous and non-humorous text. Vectors in their space, however, represent texts and position in the space is based on features of those texts. Our vector space, on the other hand, is used to dissect an individual text: separation in the space is between words of a single text rather than between two kinds of text. A more important difference, however, is that in our work, separation of texts is performed using a model of incongruity

whereas the classification used by Mihalcea and Strapparava (2005) is performed by looking for secondary features of humorous texts such as vocabulary choice or whether alliteration occurs in a text. In Mihalcea and Strapparava (2006) the authors argue that a notion of incongruity was implemented in their system - that antonymy is a simple kind of incongruity which the classifiers use, among other features, to detect humorous texts. But results suggest that this is too simplistic a notion of incongruity and the authors admit that "more sophisticated humor-specific features" need to be employed. Indeed when antonymy alone was used to separate texts, the classifiers performed only slightly better than random in discovering humorous texts: 55.7%, 51.4%, 50.5% and 51.8% accuracies resulted when trying to separate one-liners from Reuters, BNC, proverbs and commonsense statements respectively. This thesis implements a more sophisticated model of incongruity which we believe is not a secondary feature of texts but represents the crux of a certain subclass of humour.

In Mihalcea and Pulman (2007) the authors examine "the most discriminatory features learned from the text classification process" performed in Mihalcea and Strapparava (2005). When classifiers were trained, certain features emerged as important for distinguishing between humorous and non-humorous text and the authors examine these features to determine something about the nature of humorous texts. A bag of words approach was used by the classifiers and so these emergent features were words that proved to be most useful in separating humorous and non-humorous texts. The bulk of these words could be divided into two categories: words with negative polarity and human-oriented words and this offered empirical support of a hypothesis made in Mihalcea and Strapparava (2006) which claimed that these types of words appear more frequently in humorous text than in regular text. A possible improvement of the joke generator developed in this thesis might be to use, as seed texts for generation, texts that have some of the features Mihalcea and Pulman identify in this paper - i.e. texts with human oriented words and negative words might also be sought.

2.3 Register humour

2.3.1 Register

We are studying a kind of humour that involves clashes between the tone of words. As far as we know, this type of humour, which is sometimes referred to as register-based humour, has never been computationally modelled before. One reason for this might be that, as Attardo (1994) suggests, a detailed and formal definition of register has been lacking. Attardo quotes Halliday (1988), for instance, who vaguely defines register as "a cluster of associated [linguistic] features having a greater than random [...] tendency to co-occur" and Catford (1965) is similarly obscure when he defines register as "a variety correlated with the performer's social role on a given occasion". Partington (2008) does not shed much light on the matter either when he defines register as "a way of speaking or writing regularly associated by a set of participants with a certain set of contextual circumstances".

In the absence of a formal definition of register, researchers tend to develop intuitive notions of the idea. Attardo (1994) argues that register is sometimes described as 'extra' information that is somehow added to "objective descriptions of reality" and that often this added information pertains to emotion: "register deals with the affective, emotional side of language rather than with

the objective, factual aspect of communication". Other kinds of information communicated by register have been proposed such as "social roles/situations" (i.e. who the speaker is and what he/she does) and ... "field of discourse" (i.e. what the speaker is using the text for, like discussing, insulting etc)" (Attardo, 1994).

Biber (1988) makes use of quantitative measures, rather than his intuition, to examine the concept of register (which Biber refers to as 'style'). Some descriptions of register (such as Halliday's above) regard it as "a set of choices among 'linguistic features'" and Biber uses statistical methods to discover which sets of features might define particular registers. Biber is specifically interested in discovering how the registers of (transcribed) speech and writing differ. Profiles of these two kinds of text are built by computing frequency counts of 67 linguistic features (such as infinitives, wh clauses, possibility modals, and contractions) in example texts. Factor Analysis (FA) is then performed on the data to determine which features tend to co-occur in speech and which others co-occur in writing. Biber's quantitative analysis discovered, for example, that written scientific texts often contain numerous passive verbs and nominalizations but markedly few pronouns and contractions whereas the opposite occurs in conversational text. Biber suggests that these co-occurrence patterns - the expression and supression of different groups of linguistic features - can account for the different kinds of register in texts.

We are interested in estimating differences in the tone of individual words, rather than distilling the differences between speech and writing. Therefore, instead of computing the frequencies of linguistic features in texts, we compute the frequencies of words in corpora which we believe vary in lexical tone. Like Biber, we then use Principal Components Analysis (PCA) to discover the k most important dimensions of variation (k < total number of corpora). Cosine distances between the resulting word vectors are then used to estimate their difference in tone.

Paiva and Evans (2004, 2005) build on the work of Biber when they implement a natural language generator which aims to control the style of texts. Like Biber, Paiva and Evans perform a "corpus-based factor analysis" to find which groups of linguistic features determine a particular style. They then correlate these groups of linguistic features with 'internal' decisions that a natural language generator makes so that control can be exerted over the style of texts output by the generator.

Biber and Paiva and Evans, therefore, look at numerous linguistic features and attempt to determine which correlations among these features determine the styles of texts. We focus more on the register of individual words - on the variation that takes place at the lexical rather than textual level. Edmonds (1999) provides a detailed analysis of the kind of word-based register we are interested in when he develops a "computational model for representing the fine-grained meanings of near-synonyms and the differences between them". He argues that there is probably no such thing as a perfect synonym - synonyms are not identical because each one will either omit or add some connotation or nuance. The words 'fat' vs. 'Reubenesque', for example, are near-synonyms but they differ in terms of how critical and formal they are. The words 'wicked' and 'naughty' also differ in a number of ways. The word 'naughty' is less formal, has a childish overtone to it, does not have a religious connotation to it and is less forceful or intense. Edmonds refers to the differences between near-synonyms as *dimensions*. One might say, therefore, that the words 'naughty' and 'wicked' differ in terms of formality, religious, and intensity dimensions

(and some kind of adult vs. child dimension).

Edmonds identifies numerous dimensions and organizes them into categories, one of which is "stylistic tone" or register - he uses these two terms interchangeably. Stylistic tone is composed of a number of dimensions such as 'formality', 'force', 'floridity', 'familiarity', 'colour', 'concreteness', and 'simplicity'. We believe that 'formality', which Edmonds claims is "the most recognized" of the stylistic tone dimensions, is often at play in lexical register jokes, but as Edmonds notes "dimensions often cannot be defined in clear-cut terms because there are many correlations between them" (Edmonds, 1999). For example

> formal words are often flowery, forceful words are more intense or 'stronger' than others, so they are often simple, non-flowery, and concrete; floridity and colour may be the same dimension: they refer to 'bookish', literary, or sophisticated words, so they are often more formal and less simple; simple words are often more familiar or common (Edmonds, 1999).

The formality dimension therefore seems to be a kind of potpourri of different dimensions which are difficult to identify separately. Edmonds does not define a complete and discrete taxonomy of dimensions, nor does his model provide practical resources which might be of use in this thesis (such as a system that can automatically determine how a group of near-synonyms differ). But his research provides a valuable framework for understanding the different dimensions of word-based register. Interestingly, he models "a lexical-choice process that can decide which of several near-synonyms is most appropriate in any particular context" and one of the goals of our thesis is to do the opposite: to construct texts in which the most inappropriate near-synonyms are chosen, for humorous effect.

2.3.2 Automatically measuring formality

Brooke et al. (2010) implement and test various algorithms for estimating the formality of words. The simplest measure is based on word length - the longer the word, the higher its formality score (a value between -1 and 1) and another straightforward method assigns a high score to words with latinate affixes. A more complicated measure uses frequency counts of a word in "two corpora which are known to vary with respect to formality" and computes a ratio of these frequencies to estimate the formality of the word.

The most complicated algorithm in Brooke et al. (2010) uses Latent Semantic Analysis (LSA) to estimate the formality of words. A matrix in which "the row vectors correspond to the ... frequency of words in a set of texts" is computed and Singular Value Decomposition (SVD) - see Turney and Littman (2003) for details - is used to reduce the dimensionality of that matrix. Cosine distances between a word vector w, whose formality is to be estimated, and various seed words (which are known to be extremely formal or informal) are then computed and used to estimate the formality of word w.

Combining this LSA method with the other formality measures mentioned above, Brooke et al. (2010) create a system that achieves 85% accuracy in identifying "the relative formality of word pairs" when the pair of words are near-synonyms (and may not differ greatly in terms of formality) and nearly 100% accuracy when comparing words with extreme formality differences (Brooke et al., 2010).

The LSA method used by Brooke et al. (2010) is similar to one used in this thesis (and described in Chapter 3) in which frequencies of words in numerous corpora are computed and Principal Components Analysis (PCA), instead of SVD, is used to discover the k most important dimensions of variation (k < total number of corpora). Cosine distances between the resulting word vectors are then used to estimate their difference in tone.

Brooke et al. (2010) post-dates the creation of the vector space model described in this thesis (first reported in Venour et al. (2010) and then in Venour et al. (2011)) and differs from our research in some important ways. For example Brooke et al. use two lists of seed words and two corpora which are, somehow, "known to vary with respect to formality". We have deliberately decided not to deal with hard notions of formality, as this amalgam of dimensions seems hard to define. Also, we make use of 25, rather than just 2 corpora, which we think vary in terms of the kind of tone at play in lexical register jokes. Nonetheless, it would be interesting to implement and test the algorithms described in Brooke et al. (2010) in a lexical register joke classifier and generator and to compare those results with the ones yielded by this thesis.

Brooke et al. (2010) state that their LSA method was "inspired and informed" by Turney and Littman (2003) in which the authors use LSA to automatically determine the semantic 'orientation' of a word - whether a word 'sounds' positive or negative. In that same paper, Turney and Littman experiment with another statistical measure - Pointwise Mutual Information (PMI) - to infer the semantic orientation of a word and one of the classifiers built and tested in this thesis uses a variation of this method to classify texts. (See Chapter 3 for details).

2.3.3 Register-based humour

Some linguistic research on register humour has been done but Attardo (1994) argues that this research is scarce. He cites the work of Alexander who writes that register humour results from "selecting a lexeme or phraseological unit from a different style level than the context would predict" (Alexander, 1984). Alexander provides the following quote from Woody Allen as an example:

> [H]e was halfway through a new study of semantics, proving (as he so violently insisted) that sentence structure is innate but that whining is acquired (Woody Allen "Remembering Needleman" in Side Effects)

Attardo, reflecting on Alexander's example, points out that the reader of this passage does not need to understand what the terms "phrase structure" or even "semantics" mean in order to appreciate the joke - she only has to regard them as examples of "academic talk" whose tone conflicts with the informal word 'whining'. Interestingly, the classifiers and generator built and described in this thesis are more ignorant about the meaning of words than even the most obtuse reader imagined by Attardo. These systems are not equipped with any knowledge of semantics and are designed to look only to the register of a text when attempting to recognize or generate lexical register jokes. One of the aims of this thesis is to determine whether such a narrow focus can suffice in fully modelling this kind of humour.

A more recent study of register humour argues that it occurs in "diverse forms of communication" such as written literary works and semi-conversational speech (Partington, 2008). Numerous instances of register humour are found, for example, in the comic novels of P.G. Wodehouse. In

this study, Partington points out that Wodehouse's novels contain numerous informal words such as 'blighter', 'chump', 'gulped', 'bloke' and 'baffled' as well as "strikingly formal" words and phrases like "injudicious", "at this juncture", "the work of a moment" and "paltering with the truth". Partington notices that certain phrases are often repeatedly invoked to mix "formalisms and informalisms very closely in the same segment of text" (Partington, 2008). For example the formal phrases "as regards" and "endeavouring to" are frequently combined with informal words or phrases:

> as regards his getting blotto
>
> as regards the fusing of her soul and mine, therefore, nothing doing
>
> ... where Bill, the fox-terrier, had encountered an acquaintance, and, to the accompaniment of a loud, gargling noise, was endeavouring to bite his head off

Partington points out other kinds of register humour in Wodehouse's writing. For example "banal, mundane events" are often expressed using "high-flown language":

> Mr. Fink-Nottle appears to have realized at this point that his position as regards the cabman had become equivocal. The figures on the clock had already reached a substantial sum, and he was not in a position to meet his obligations.

The narrations of Wooster, a recurring character in the novels, demonstrate "constant movement between the elevated and the mundane" and this same kind of movement can also be seen in exchanges between characters. Indeed in the following dialogue, the characters self-consciously allude to the change in register:

> I was able to insert a chemical substance in his beverage which had the effect of rendering him temporarily insensible.
>
> You mean you slipped him a Mickey Finn?
>
> I believe that is what they are termed in the argot, madam.

Partington's examples demonstrate that register play can occur at the level of words, phrases, syntax and subject matter. This thesis focuses exclusively on word-based clashes of register, however, but the other ways in which incongruities of tone can be expressed would be interesting topics for future research.

Chapter 3

Estimating difference in tone

3.1 Our goal

Clashes of a certain kind of tone occur in lexical register jokes. We are interested in building a semantic space in which position reflects a certain kind of tone because we believe that patterns unique to lexical register jokes will be seen in such a space. If the text of the Simpsons joke shown in Section 1.3, for example, were plotted in this space, words like 'Superbowl', 'gee', 'fellas', and 'gonna' would cluster together because they all have a relatively informal tone, whereas the more formal sounding word 'crestfallen' would appear as an outlier. This text is an example of the simplest kind of lexical register joke in which the tone of a single word is incongruous with the tone of the rest of the text. In more complicated lexical register jokes, groups of words are in opposition to each other, and we would expect these opposing sets of words to appear as different clusters in the space.

This chapter describes our first attempts at building a semantic space which will reflect the tone of words and exhibit such distinctive patterns when words from lexical register jokes are plotted into it. We believe that a space which estimates how words score in terms of a certain kind of tone:

- will confirm our hypothesis about the tonal structure of lexical register jokes
- can be used to automatically distinguish between lexical register jokes and regular text
- can be used to generate new lexical register jokes.

Most importantly, a semantic space in which position reflects a certain kind of tone would make the idea of "clashes of tone" more precise and would provide an objective and quantifiable way of measuring a certain kind of humorous incongruity - a concept which, from the literature we have seen, has proven hard to measure or even define.

3.2 Why not use a lexicon?

We believe that there is a significant subset of lexical register jokes in which the relevant dimensions of opposition have something to do with formality, archaism, literariness, etc. (For the sake of brevity, we will allude to this cluster of features as tone). Initially we hoped to find a lexicon or dictionary which might help us grade words in terms of these dimensions. Using this resource, we could then detect differences in the tone of words and if this difference were to exceed some

empirically determined benchmark, a program could conclude with some certainty that the text is a lexical register joke.

We looked into using machine readable resources such as the American Language Standardized Dictionary[1], the Moby Thesaurus[2] or an electronic version of Roget's 1911 thesaurus[3] but found that they do not provide information about the kind of tone we are interested in. In fact these lexicons do not even provide partially useful information such as whether a word is archaic or not.

Other resources such as WordNet, the MRC psycholinguistic database and the lexicon created for the STANDUP project (Manurung et al. 2008) provide familiarity ratings and we felt that this might be potentially useful information because familiarity may have something to do with our vague cluster of concepts (i.e. formality, archaism, literariness, etc). Even so, familiarity is (possibly) only one of a number of features we are interested in, so the search for resources continued.

The General Inquirer lexicon[4] contains 11,788 words which are tagged in terms of 182 categories. The lexicon marks, for example, whether a word is positive, negative, strong or weak, and whether a word has certain overtones such as connotations of virtue, vice or pain for example. Unfortunately none of the GIL categories provides information about how a word scores in terms of the tone we feel is at play in lexical register jokes. Also, the GIL only provides, for the most part, 2 point scales (e.g. a word either has an ethical overtone or not). We are interested, however, in knowing not only whether words have a certain kind of tone but to what degree they express this tone.

From these investigations, it appeared that existing lexical resources probably contain little of the kind of information we require. The search for resources was hampered also because it is difficult to create precise definitions of the dimensions at play in lexical register jokes, nor do we know which of them are independent of each other, nor exactly what combination of these dimensions is involved in humorous incongruity. It was decided then that a way of specifying the property we are interested in might be to make use of example corpora that seem to vary on it.

3.3 Creating a semantic space

In the absence of lexicons which might estimate how individual words score in terms of the vague and poorly defined amalgam of dimensions at play in lexical register jokes, a statistical approach which uses methods commonly employed in information retrieval was adopted. 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

[1] http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/corpora/dicts/sigurd/0.html
[2] http://icon.shef.ac.uk/Moby/
[3] from Project Gutenberg http://www.gutenberg.org/wiki/Main_Page
[4] available at http://www.wjh.harvard.edu/~inquirer/

- plot words from texts into the space
- try various outlier detection methods to see which one displays the outlier and clustering patterns we anticipate seeing.

It is important to note that a classifier making use of this semantic space will only attempt to estimate whether there is uniformity or disparity of lexical tone in a text. It indicates for instance if the words of a text are relatively close together in the vector space, but what kind of tone these words uniformly possess - whether they are all quite formal, informal or neutral - remains unknown. In other words the vector space classifier will simply regard a text as a set of vectors that inhabit a vector space and determine how close or how far apart they are from each other in the space. It does not know which parts of the space tend to contain formal words, which area is the neutral zone or which is the informality subspace.

It is also important to note that in the chapters that follow (Chapters 3-5), the semantic space is being developed and fine-tuned in order to maximize its accuracy in distinguishing between the lexical register jokes and regular texts of the development set. In this development stage, scores are used to guide design decisions and determine the values of certain parameters of the semantic space. Once this formative work is done, our model will be formally tested (in Chapter 6) on unseen joke data.

3.3.1 Assumptions of our model

Our methodology for creating a feature space, the contours of which we hope will allow empirical measurement of words in terms of our ill-defined cluster of concepts, is based on three assumptions:

1. that the corpora actually display differing degrees of formality, archaicness etc. – dimensions which we think are at play in lexical register jokes.

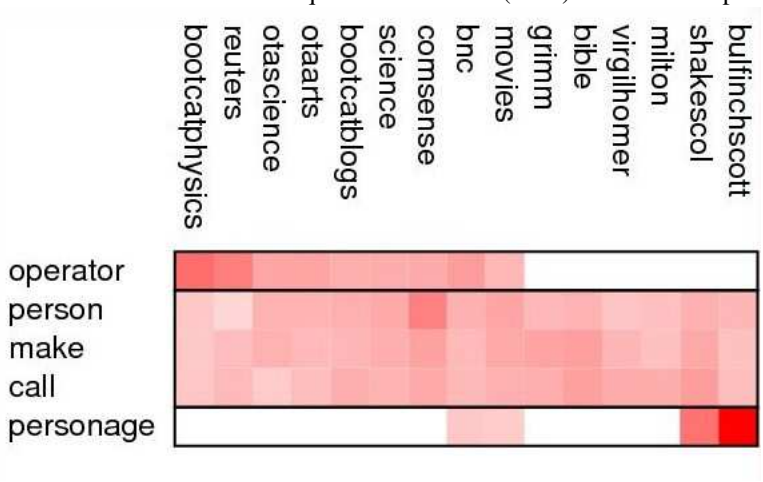
2. that word choice is a significant determiner of tone in a text. Syntax and metaphor, for example, may also play a very important role, but these are not considered here. This will be truer for some corpora than for others however. For example the Shakespeare corpus has many archaic words in it (e.g. ‘doth’, ‘beseech’, ‘thou’) but in Jane Austen’s novels, for instance, formality may be expressed not as much through individual words but through syntax, metaphor, etc.

3. that frequency counts of a word in the various corpora can act as a kind of “tonal fingerprint” of that word. For example, in Figure 3.1 the 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 would be considered similar in tone. Whereas the pattern for ‘personage’ is quite different, indicating that its tone may be different.

3.3.2 The corpora

We looked for corpora which we think display differing degrees of degrees of formality/archaism/literariness – corpora such as the King James version of the bible, Virgil’s The Aeneid, all of Shakespeare’s

Figure 3.1: Using frequency count patterns as "tonal fingerprint". Cells in the table represent the frequencies of words (rows) in various corpora (columns).



plays and then more modern collections such as a year's worth of New Scientist articles, newspaper articles from Reuter's and a collection of common sense statements. 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. Table 3.1 shows the set of 12 corpora used in the initial set of experiments.

3.4 Automatically identifying an incongruous word

Twenty lexical register jokes were used to develop the model. All contained exactly one word (shown in bold in Tables 3.2 and 3.3) whose tone we judged to be incongruous with the rest of the text.

Most of the jokes (15/20) are captions taken from cartoons appearing in the New Yorker magazine. Joke #10 however is taken from a scene in Monty Python's 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.

Twenty-two additional lexical register jokes, also captions taken from New Yorker cartoons, were reserved for the test set. These will be discussed in more detail in Chapter 6.

3.4.1 Computing scores

In the tests, stopwords were filtered from a lexical register 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

Table 3.1: Corpora A

no.	corpus	no. of words	source
1	Virgil's The Aeneid (17th century translation)	108,677	Oxford Text Archive (OTA)
2	all of Jane Austen's novels (early 19th century)	745,926	OTA
3	King James bible (17th century translation)	852,313	Project Gutenberg
4	All of Shakespeare's tragedies and comedies (1623 first folio edition)	996,280	OTA
5	Grimm's fairy tales (19th century)	281,451	Project Gutenberg
6	All the poems of Samuel Taylor Coleridge (early 19th century)	101,034	OTA
7	Two novels by Henry Fielding (18th century)	148,337	OTA
8	Collection of common sense statements	2,215,652	Open Mind Common Sense http://commons.media.mit.edu
9	Corpus of Reuter's new articles	1,614,077	nltk package
10	New Scientist articles	366,393	nltk package
11	Movie reviews	1,298,728	nltk package
12	the written section of the British National Corpus (World Edition).	80 million	Frequency counts of a word in the BNC were taken from the CUVPlus dictionary, available at the Oxford Text Archive.

Table 3.2: Development set of lexical register jokes

no.	lexical register joke	source
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	Mom Bart is on a strict diet of complex carbohydrates steak will make him **logy**.	The Simpsons, "Dead Putting Society"
3	Thirty two years as a chef and he can still go **yum**.	CCNY 1970, p.90
4	Cancel my appointments for this afternoon miss I am **rapping** with my son.	CCNY 1970, p.582
5	Gentlemen nothing stands in the way of a final accord except that management wants profit maximization and the union wants more **moola**.	CCNY 1970, p.596
6	Sticks and stones may break my bones but **rhetoric** will never hurt me.	CCNY 1970, p.624
7	Well all our data confirm your own original diagnosis mrs you are just plain **tuckered** out.	CCNY 1971, p.246
8	Sometimes I think you are a serious research and development man and sometimes I think you are just **messing** around.	CCNY 1971, p.753
9	When they recommend serving it at room temperature they are referring of course to the rooms of **yesteryear**.	CCNY 1973, p.295
10	You cannot expect to wield supreme executive power just because some watery **tart** threw a sword at you.	Monty Python and the Holy Grail
11	Oh, how could I fall for fake (Superbowl) tickets? Gee, the fellas are gonna be **crestfallen**.	The Simpsons, "Sunday Cruddy Sunday".
12	Friends we have temporarily lost the video portion of our talk show but will continue to bring you the inane **flummery** of our panelists.	CCNY 1970, p.620

Table 3.3: Development set of lexical register jokes (continued)

no.	lexical register joke	source
13	Tell me compared to your other victims how would you rate me on **sangfroid**.	CCNY 1970, p.85
14	**Gee** Mr. Determining which issues have growth potential while simultaneously working to provide your clients with a reasonable annual yield is most certainly creative.	CCNY 1974, p.535
15	It is best not to use big words. Why choose a big word when a **diminutive** one will do?	joke attributed to W.C. Fields
16	Please can I have the thirty cents this week without the **jawboning**?	CCNY 1971, p.186
17	Damn it, agree to whatever she demands no matter what it takes I want my **mommy**.	The New Yorker July 18, 1994
18	Listen serving the customer is **merriment** enough for me.	The Simpsons, "Twenty-Two Short Films About Springfield"
19	Last chance to see the beautiful reflections in Mirror Lake before **eutrophication** sets in.	CCNY 1971, p.569
20	The market gave a good account of itself today **daddy** after some midmorning profit taking.	CCNY 1972, p.195

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 order 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 register jokes was computed to be 9.7 out of a maximum of 40.

3.4.2 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 (Li and King, 1999), 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 (Li and King, 1999) 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 (Salton and McGill, 1986) and was used here to produce a ranked list of words in the manner described in 1. above[5].

3.4.3 Initial results

Table 3.4 shows the outcomes of testing on development examples using the set of corpora A (listed in Table 3.1). 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 3.4: 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

3.4.4 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 (Salton and Buckley, 1988) 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

[5]The cosine of two vectors u, v is defined as their dot product divided by the product of their lengths. When u or v is a zero vector, the result is 0/0. When faced with having to compute the cosine distance between a zero vector and another vector, we have elected to have the classifier output a cosine distance of 0. See Section 4.3.3 for details.

Table 3.5: Frequency count matrix for content words in joke #1

word	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12
operator	33.4	17	5.3	8.1	9.0	0	0	0	0	0	0	0
make	365.5	767	1225.8	988.0	3313.6	742.5	742.5	1238.9	432.4	1067.1	957.2	663.1
personage	0	1	0.7	0	0	0	0	0	0	4.0	6.7	9.8
person	14.8	254	251	251.0	3863.4	46.1	46.1	65.7	46	348.5	647.1	29.6
call	116.4	173	161.6	144.6	314.1	95.9	95.9	228.7	266.8	223.8	330.3	386

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 **X** where the value of the cell in row i and column j is the normalized frequency count of word i in corpus j. For example Table 3.5 shows the frequency count matrix for the content words of joke #1. Our second method of pre-processing, which has "been found to be very helpful in information retrieval" (Turney, 2006), involved computing the log entropy of the columns of matrix **X**. This amounts to giving more weight to columns (i.e. corpora) that are better at distinguishing rows (i.e. words). Let x_{ij} be the cell in row i and column j of the frequency count matrix **X** which has m rows and n columns. Our aim is to weight the cell x_{ij} by the entropy of the j-th column. Turney (2006) describes how to do this:

   "To calculate the entropy of the column, we need to convert the column into a vector of probabilities. Let p_{ij} be the probability of x_{ij}, calculated by normalizing the column vector so that the sum of the elements is one, $p_{ij} = \frac{x_{ij}}{\sum_{k=1}^{m} x_{kj}}$. The entropy of the j-th column is then $H_j = -\sum_{k=1}^{m} p_{kj} \log\left(p_{kj}\right)$... We want to give more weight to columns ... with frequencies that vary substantially from one row ... to the next, and less weight to columns that are uniform. Therefore we weight the cell x_{ij} by $w_j = 1 - H_j / \log(m)$, which varies from 0 when p_{ij} is uniform to 1 when entropy is minimal.We also apply the log transformation to frequencies, $\log(x_{ij} + 1)$. (entropy is calculated with the original frequency values, before the log transformation is applied). For all i and j, replace the original value x_{ij} in **X** by the new value $w_{ij} \log(x_{ij} + 1)$" (Turney, 2006).

Tf-idf transformations (Table 3.6) 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.

3.4.5 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 incorporate these methods as permanent features of the system and subsequent experiments focused on varying the corpora used to compute frequency counts. Table 3.7 show the results of this experimenting. Information about the different corpora

Table 3.6: 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.

Table 3.7: Results from using different sets of corpora

Test no.	Pre-processing	Distance metric	Corpora	Score (out of 40)
1	log entropy	cosine	B	31
2	log entropy	cosine	C	35
3	log entropy	cosine	D	37

sets (B, C, D) we used - details of the kind shown in Table 3.1 - are provided in the tables in Appendix A.

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 British university students[6], a corpus of informal blogs, and a corpus of documents about physics[7]. As we see from Table 3.7, 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.

The decisions made in constructing corpus 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. Individual words in our model can be compared to atoms which crystallographers use to bombard crystals in order to determine the shape of those crystals. In our case however, we are 'bouncing'

[6]The latter two corpora are from OTA

[7]The latter two corpora were created using SketchEngine http://www.sketchengine.co.uk/

words off of corpora, our targets, not to determine characteristics of the corpora, but characteristics of the words themselves. To differentiate between words in ways we care about requires experimenting with different targets. 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: 92.5% of the possible maximum score in identifying the incongruous words in the development set of lexical register 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 register joke i.e. by plotting them, along with content words, into the space. (Recall that we are fine-tuning our algorithm here using development data. Once this formative work has been done, our model will be tested on unseen lexical register joke data). Incongruous words in lexical register 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.

3.4.6 What kinds of incongruity were detected?

Our method seems able to label as incongruous, the following types of words:

- words that are archaic compared to other words in a text: e.g. 'yesteryear', 'crestfallen', 'merriment'

- words that are 'fancier' than other words in a text: e.g. 'rhetoric', 'logy', 'personage', 'sangfroid', 'diminutive'('flummery' was also considered incongruous but only because it did not appear in any corpora and so had a string of zeroes as its frequency count pattern).

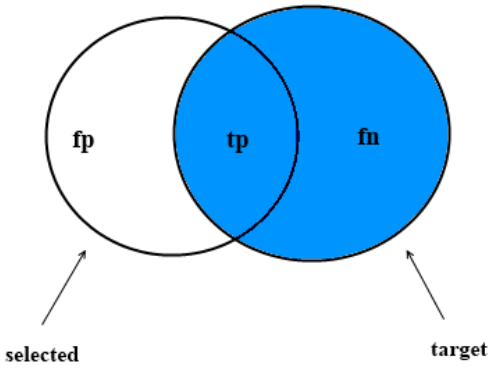
- words that are less 'fancy' than other words in a text: e.g. 'rapping', 'yum', 'messing', 'tart', 'gee', 'mommy', ('tuckered' and 'moola' were also considered incongruous in their respective texts but, again, only because they did not appear in any of the corpora).

The semantic space failed to identify the most incongruous word in only one case: in lexical register joke #20, the word 'midmorning' was chosen instead of 'daddy' (which appeared second in the ranked list). This mistake probably occurred because the word 'midmorning' appears in only one corpus. It may have been identified as unique, not so much because its tone is truly discordant with the other words but because the corpora data is simply too sparse. (The problem of corpus sparsity is discussed in detail in Chapter 4). Also the word 'midmorning' is sometimes spelled 'mid-morning', making the frequency counts for 'midmorning' even lower than they might have been if there were no alternative spellings.

3.5 Automatically distinguishing between lexical register jokes and 'regular' text

The next step is to determine whether the space can be used to detect lexical register 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 exceed a threshold, the program would predict that the text is a lexical register joke.

Figure 3.2: Computing precision and recall



This approach was tested on a set of texts consisting of the development set of lexical register jokes together with a sample of 'regular' i.e. non lexical register joke texts: newspaper texts randomly[8] selected from the June 5 2009 issue of the Globe and Mail, a Canadian national newspaper (see Table 3.8). Complete sentences from the newspaper were initially much longer than the lexical register joke sentences - the average number of words in the lexical register 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.

3.5.1 Measuring the algorithm's performance

There are a number of ways of quantifying the algorithm's success in distinguishing between lexical register jokes and 'regular' text. To simplify somewhat, there are two possible perspectives.

3.5.1.1 Retrieval

In such a task, the program has to select a relatively small number of target items within a much larger set of items, as when searching for a relevant document in a library or a matching page on the world-wide web. Here there are two measures of success: to what extent did the search program miss items it should have found, and to what extent were the items it claimed to have found actually correct? The first of these is measured by recall which, based on Figure 3.2, is $recall = \frac{tp}{\|target\|} = \frac{tp}{tp+fn}$. In the figure, tp means true positive, fp means false positive and fn means false negative[9].

The second measure of success is precision which, using the variables in Figure 3.2, is the ratio $precision = \frac{tp}{\|selected\|} = \frac{tp}{tp+fp}$. The nearer a ratio is to 1.0 the better. Notice that there is a trade-off here: a program could achieve a perfect recall score of 1.0 by simply choosing every item

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

[9]The figure is taken from Manning and Schutze (1999)

Table 3.8: Phrases randomly selected from a newspaper

no.	newspaper text
1	the tide of job losses washing across north america is showing signs of ebbing, feeding hope that
2	yet investors and economists are looking past the grim tallies and focusing on subtle details that suggest
3	there is definitely dancing at the prom, an annual rite for muslim teens but no boys, no
4	it is a shining example of the intersection of cultures that the president stressed in his historic
5	he insisted last night that he would not waver or walk away from power at the end
6	members of both are forever claiming public interest and good as the end goal while rarely declaring
7	it was also pointed out that some of the ministers with the most controversial expenses claims, details
8	almost always the only completely opaque cost is legal aid. Lawyers oppose such disclosure
9	the chance to apply for early parole after serving so many years behind bars would be to deny
10	a panel of international experts is recommending that the way diabetes is diagnosed should be dramatically simplified
11	bumping into a member of the british royal family was an unexpected bonus for the visitors who
12	even if the resolution does not pass, it has succeeded she said in shining a spotlight on
13	the resolution reflects growing dismay among municipalities over being shut out of lucrative infrastructure jobs as a
14	one of the dominant phenomena in the art world of the past three decades or so has
15	the exhibition is a historic event formally linking at last the planet's two leading centres for the
16	at thursday night's performance the boy beside me who could not have been more than eighteen actually shrieked in
17	both runs were completely sold out and he was so mobbed at the stage door that he
18	the bid has already been won. what is to stop them from squeezing the architect for cuts to
19	advertising executives packed into the hall for a television presentation, a rite of spring passage in the world
20	he contorts a bit raising his right shoulder wringing one hand with the other and fingering his

in the entire collection, but its precision would be extremely poor. Sometimes precision and recall are combined into a single F-score, to allow for potential trade-offs.

3.5.1.2 Classification

In such a task, the program has to classify all the items in the supplied collection into some small set of categories, often just two. In this, the central success measure is how many times the program's choice of category was correct. This is usually computed as accuracy: (number of correct classifications)/(number of classifications made). These measures are sometimes described as values between 0.0 and 1.0, sometimes as percentages.

If we had mixed our 20 joke texts into a very large number of other texts, and set our program to 'retrieve' just the jokes, then measures of precision and recall would be appropriate. However, our data set was evenly balanced between the two types of text, which means that these measures tell us less. This is largely because the relationship between precision and recall is undermined, and a program could achieve not only 100% recall but 50% precision simply by retrieving all 40 items. This gives an F-score of 66%, which is not bad for a completely dumb program.

Classification tasks, on the other hand, often operate on balanced data, and are routinely assessed in terms of accuracy, which is not distorted by this mix of data. We therefore regard accuracy as the primary success measure in these tests.

As a secondary measure of success, however, precision scores were also calculated. This is because we are interested in developing an algorithm which will lead to the generation of lexical register jokes. For that purpose, not all forms of inaccuracy are equally bad. For a generator, it is desirable that as many as possible of its output items are indeed what they should be (jokes); it matters much less if it is tacitly overlooking chances to generate other possible jokes. Hence precision (the percentage of items which the computation deems to be jokes which really are jokes) is more pertinent than recall (the percentage of the possible jokes which the computation manages to identify).

3.5.2 Classification results

Accuracy is highest when the threshold cosine value (which we will refer to as the "joke boundary") is arbitrarily set at 0.5 - i.e. when we say that λ needs to be less than or equal to 0.5 in order for the text to be considered a lexical register joke.

Recall that a lower cosine score means a higher distance in the space. This can cause confusion when discussing measurements in the vector space and so to minimize the confusion, we will use the term 'similarity' rather than 'distance' when comparing the position of word vectors in the vector space.

Thus, to be clear, two word vectors that are far apart in the space form a low cosine value. The similarity of these two words is therefore low and if that similarity value falls unders the "joke boundary", the text containing them is considered a lexical register joke.

From Table 3.9 we see that 80% accuracy (in detecting jokes from within the set of all the texts processed), 73.1% precision and 95% recall result using this "joke boundary". (When pathological cases[10] are excluded from the evaluation, the program achieves 28/35 (80%) accuracy, 15/21 (71%) precision and 15/16 (93.8%) recall using this threshold).

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

Table 3.9: Accuracy, precision and recall when computing averages

joke boundary	accuracy	precision	recall	F score
<=0.5	32/40 (80%)	19/26 (73.1%)	19/20 (95%)	82.6
<=.425	30/40 (75%)	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 register joke, but these results show that it has limited success in estimating how incongruous a word is. We believe that differences in tone in lexical register jokes are much larger than those in regular text but the semantic space achieves, at best, only 80% accuracy 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 "joke boundary" is set at 0.5, the three newspaper texts (not containing a pathological word) mistakenly classified as lexical register 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.

Or perhaps the problem is more deeply rooted. New sentences which no one has ever seen before are constructed every day because writing is creative: when it is interesting and not cliched 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 at gauging how incongruous they are because distinguishing between innovative word choice and incongruous word choice is currently beyond its reach.

3.6 Improving the detection of lexical register jokes

This section discusses a method which attempts to improve the detection of lexical register jokes. This methods aim to do so, not by addressing the innovation vs. incongruity problem mentioned above[11], but by using Principal Components Analysis (PCA) to refine the feature space which has been built so far.

3.6.1 Using PCA to improve estimates of tonal difference

Our first attempt at improving the system involves using PCA to construct a smaller coordinate system in which estimates of tone might be enhanced. PCA is a popular method "used abundantly in all forms of analysis – from neuroscience to computer graphics" which can "reduce a complex

[11] the solution of which is outside the scope of this thesis

data set to a lower dimension to reveal the sometimes hidden, simplified structures that often underlie it" (Shlens, 2009).

PCA will find, for example, that when scores are relatively high in corpus 1, they are also relatively high in corpora 3 and 8. Frequency counts in corpora 1, 3 and 8 are considered correlated if this pattern repeats a statistically significant number of times. In such a case, the three features, which appear as three separate dimensions in the feature space, can be collapsed into a single dimension or component, thereby simplifying the space and making its dimensions more orthogonal.

More importantly, the reduced space can potentially provide more accurate estimates of how objects score in terms of a certain property. Our hope is that PCA will find components (i.e. groups of correlated features) in the feature space which have something to do with the tone of words. Distances between words could then be measured in terms of only those components rather than in terms of all the features of the space. In this way features which do not have a bearing on tone will receive lower weightings when distance measurements are calculated, making these distance measurements more precise. Improvements in this regard will in turn enhance the system's precision in differentiating between lexical register jokes and regular text.

3.6.2 Why PCA and not FA?

Factor Analysis (FA) divides variance into three types:

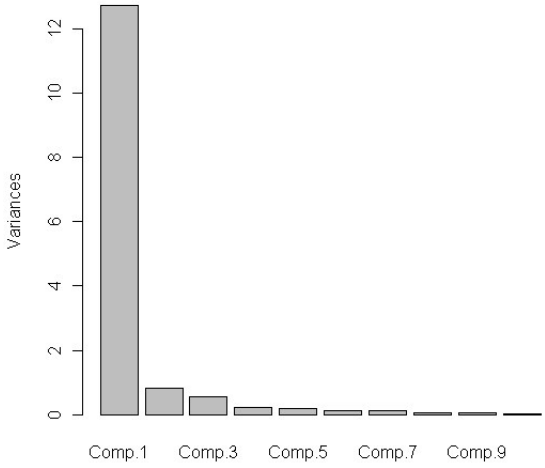
1. Common variance: variance of a variable that is shared with other variables in the analysis.
2. Specific variance: variance specific to a variable.
3. Error variance: variance introduced in a variable by noise or error in the data-gathering process Paiva 2004.

In FA, "factors are extracted only from the common variance" (Paiva, 2004). PCA, however, treats variance as a "uniform object" - it does not ignore specific and error variance when determining underlying factors in the data. Whether one uses FA or PCA sometimes depends on the nature of the data to be analysed.

For example if we were investigating whether age affects a man's reflexes, we might see how men in three different age groups - say 20 year olds, 40 year olds and 60 year olds - score in a reflex test. Let's say the data is stored in a matrix – the first column contains the scores of the 20 year olds, the second column holds all the 40 year olds' test scores and the third column, the 60 year olds' scores.

In a Factor Analysis, the variance within each age group/column would be excluded and only the variance between the columns of the matrix would be used for extracting factors. Thus all the 20 year olds, for example, would be regarded as essentially the same, in terms of their reflexes, if a Factor Analysis were performed here. Making this simplifying assumption is probably not a mistake in the context of this experiment, but it would be a serious error in the case of our experiments. The columns of our frequency count matrices represent corpora and the rows consist of words that appear in a given text (which may or may not be a lexical register joke). Treating those words as roughly identical, like the objectified 20 year olds in the reflex experiment for example, would be a serious mistake as these words are not roughly equal in terms of their frequencies within a

Figure 3.3: Results of PCA on matrix **X** (i.e. stopwords appear in the 14000 word sample and no log entropy pre-processing was performed).



corpus. In other words, the variance of a variable (i.e. corpus in our case) is not something that can be ignored when searching for correlations in the frequency count data. For this reason Principal Components Analysis, which considers all variance in data as relevant, was chosen.

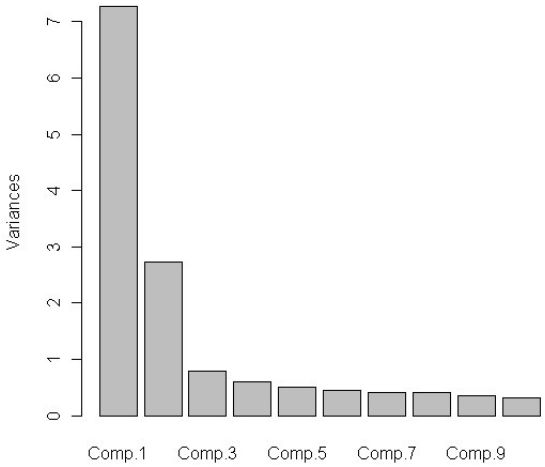
3.6.3 Performing PCA

Currently the feature space which performs best in estimating differences of tone has 15 features: each feature is a word's frequency count in one of the 15 corpora of corpus set D. To perform PCA a large sample of words was created by randomly choosing 1000 words from each corpus in corpus set D, except the BNC, resulting in a list of 14,000 words. (Words were not randomly selected from the BNC because our system does not make direct use of this corpus. Instead it uses the CUVPlus dictionary to obtain frequency counts of words in the BNC. Therefore 1000 words were selected from each of the corpora in corpus set D except for the BNC). Frequency counts of these 14000 words in the 15 corpora were then computed, resulting in a 14000 x 15 matrix (matrix **X**). PCA was performed on the matrix to determine whether it is simply an undifferentiated mass - a kind of numerical Pangaea - or whether it can be divided into component 'continents'.

PCA usually performs a pre-processing step which standardizes the scores of each column of the matrix to a mean of 0 and a standard deviation of 1 - making the variance of each column equal to 1 (Hinton, 2004; Manly, 2004). Since there are 15 columns in matrix **X**, the total variance to be accounted for will be 15.

When PCA was performed on matrix **X**, numerous components were found but as the screeplot shown in Figure 3.3 suggests, the first component accounts for most of the variance in the data. The y axis of the graph shows how much of the total variance is 'explained' by a component and we see that component #1 accounts for more than 12/15 of the variance - the exact figure is 84.7%. Looking at the loadings of this component (a loading is a measure of how much each feature contributes to a component), we find that the loadings for each of the features are nearly equal. Since each feature contributes equally to this first component and the component accounts for most of the variation in the data, the PCA results suggest that all the corpora are highly and positively correlated.

Figure 3.4: Results of PCA on matrix K (i.e. stopwords are excluded from a new 14000 word sample and log entropy of the columns performed)



We wondered, however, whether the PCA had failed to find more than one significant component in the data because stopwords were included in the sample. Including stopwords might obscure important differences between the corpora and make them seem more similar than they are. In other words, we want to find corpora that are correlated for reasons other than they share the same stopwords.

A new sample of words which does not contain stopwords was therefore compiled. 1000 content words were randomly chosen from each of the corpora, resulting in a list of 14,000 words. Proper names and abbreviations were then removed from the list for the opposite reason stopwords were excluded. Proper names and abbreviations rarely appear in more than one corpus - the name 'Shadrach', for example, appears in the bible but is unlikely to appear in any of the other corpora - and the concern was that if too many of these 'single-corpus' words were to appear in the sample, the PCA might make the opposite (but equally uninteresting) conclusion that each corpus is unique. After removing these kinds of words by hand, a list of 13060 words remained and, as before, frequency counts of these words were computed in the 15 corpora, resulting in a 13060 x 15 matrix (matrix **J**).

Given that the first PCA failed to find interesting patterns in the data, log entropy preprocessing of matrix **J** was performed for this second PCA. This step was taken for the same reason stopwords were excluded: to underscore non-trivial correlations and differences between the corpora, with the hope of improving the acuity of the Principal Components Analysis.

A total of 15 components were suggested by the second PCA but the first three, as we see from Figure 3.4, are the most significant. The first component explains 48.4% of the variation, the second component 18.1% and the third 5.3% - together they represent 71.8% of the total variance of the data.

As in the first PCA, the loadings of component #1 are roughly equal (see column 1 Table 3.10) and so this first component consists of all 15 features.

The second component consists of the group [Shakespeare/Coleridge, Virgil/Homer, Milton, the Bible, Bulfinch/Scott and Grimm's fairy tales] and the inversely related group [OTA science,

Table 3.10: Components resulting from 2nd PCA. The numbers in the columns are feature loadings.

Corpus number and name	Component 1	Component 2	Component 3
1 Virgil/Homer	-0.2356	-0.3468	-0.1474
2 The bible	-0.2008	-0.3420	0.1138
3 BNC	-0.3349	0.1418	-0.0212
4 Common sense	-0.2907	0.0962	0.3461
5 Grimm's Fairy Tales	-0.2354	-0.2866	0.2216
6 Movie reviews	-0.3053	0.0628	0.3442
7 Reuters	-0.2451	0.2349	-0.2660
8 New Scientist	-0.2847	0.2344	-0.0293
9 Shakespeare/Coleridge	-0.2005	-0.3494	-0.0659
10 Bulfinch/Scott	-0.2594	-0.2924	-0.1363
11 Milton	-0.2046	-0.3426	-0.2531
12 Physics (via bootcat)	-0.2514	0.2376	-0.1446
13 OTA science	-0.25705	0.2862	-0.2490
14 Blogs (via bootcat)	-0.2689	0.0458	0.5889
15 OTA arts	-0.2587	0.2679	-0.2988

Table 3.11: Words receiving highest and lowest component #2 scores

words with highest C2 scores	words with lowest C2 scores
system	thou
based	thy
research	thee
development	shall
data	hast
problems	ye
problem	shalt
international	behold
group	thine
technology	lord

OTA arts, Physics, Reuters, New Scientist, BNC][12]. The first group consists of mostly archaic works of literature while the second group is comprised of modern works and so component #2 could be regarded as a dimension that estimates how archaic a word is. Words which receive the highest and lowest scores for this component are shown in Table 3.11 and they provide further evidence that component #2 could be interpreted as a kind of archaic vs. modern dimension. It is rather remarkable to find a component apparently related to something as subtle (and semantic) as tone, lying latent in something as simple (and bloodless) as frequency count data. This offers further encouragement that assumption #3 of our model (described in Section 3.3.1) may not be as naive as one might have feared.

The third component consists of the group [blogs, common sense, movie reviews, Grimm's

[12]The loadings for the blogs, movie reviews and common sense statements are low, indicating that they make a negligible contribution to the component and so they do not appear in either group.

Table 3.12: Words receiving highest and lowest component #3 scores

words with highest C3 scores	words with lowest C3 scores
movie	labour
fun	favour
shit	region
girls	increase
stupid	yield
funny	greater
girl	proposed
cute	regions
favorite	cited
movies	relation

fairy tales, the Bible] and the inversely related group [OTA arts, Reuters, Milton, OTA science, Virgil/Homer, Physics, Bulfinch/Scott][13]. Words which receive the highest and lowest scores in terms of component #3 appear in Table 3.12. Words with the highest scores are generally common, modern, and informal sounding. Words with the lowest component #3 scores also seem commonplace, although they are neither particularly modern or archaic sounding. Also, they are relatively more formal and 'serious' but not extremely so - they are not particularly learned or arcane, for example. Therefore component #3 might be viewed as a (perhaps flawed) 'common-modern-informal' vs. 'common-modern-formal' dimension. The extent to which this component is useful will become apparent in the experiments (described below) in which it and the first two components are used to measure incongruities of tone.

The 12 remaining components suggested by the PCA only account for 18.2% of the variance in the data and so were not used. (It is standard practise in PCA to "discard any components that account for only a small proportion of the variation in the data" (Manly, 2004)). Nor did the remaining components seem related to tone. It is difficult to imagine, for example, what kind of tone might be characterized by the following group of dissimilar corpora: works by Milton, essays on physics, blogs, movie reviews and Shakespeare's plays. These corpora are correlated, however, in component #4 and components #5 - #15 propose other correlations which are equally discordant in terms of tone.

Thus only the first three components suggested by the PCA account for most of the variance in the data and have something to do with tone. Only these will be used to create a new semantic space in which we hope estimates of incongruity are improved. Because all 15 features of component #1 are positively correlated and have similar loadings, the 15 dimensional space which yielded the best results in previous testing (described in section 3.5) is essentially being collapsed into a single dimension in the new space. If words were scored along this dimension alone, we would expect similar results to those yielded by the 15 dimensional space: a score of 39 out of a maximum of 40 in identifying incongruous words in a text and, at best, 80% accuracy in identifying lexical register jokes in a set of texts. The new space, however, incorporates components 2 and 3 and we are hoping that they will act as refinements and improve the system's estimation of

[13]The BNC, New Scientist and Shakespeare/Coleridge corpora have low loadings i.e. contribute little to this component and so do not appear in the groups.

tonal difference.

3.6.4 Creating the reduced space

The following is a detailed list of steps that were taken to create the reduced space:

1. create a list of 14000 content words (i.e. exclude stopwords).
2. remove abbreviations and proper names, resulting in a list of 13060 words.
3. compute frequency counts (per million) of these words in corpus set D. This results in a 13060 x 15 matrix. This represents a sample population – a sample of English words from which we will get an idea of what the distribution of frequency scores and their resultant component scores looks like.
4. compute the log entropy version of these scores.
5. perform PCA on the above matrix. Various components emerged. The first three components and various subsets of these components will be used to build semantic spaces in which words are plotted and estimates of tonal incongruity are made.
6. compute the first three component scores for the log entropy version of the sample population. This results in a 13060 x 3 matrix.
7. compute the means and standard deviations of the columns of the above 13060 x 3 matrix.

3.6.5 Using PCA subspaces for classification

To plot the words of a text such as a lexical register joke into the component space (or a subspace), the following steps are taken:

1. compute the frequencies (per million) of the content words in a lexical register joke (for example “sticks stones break bones rhetoric hurt”).
2. perform log entropy transformation of these frequencies.
3. compute the component scores for the words in the lexical register joke.
4. convert these component scores into zscores. To do this we use the information from step #7 above. (Each component will have a different range of scores so to make the axes or dimensions of the component space comparable, we standardise the scores. We were uncertain whether taking this step is necessary however and Section 3.6.6 shows the results of testing the PCA subspaces when this conversion into zscores is not performed).
5. compute cosines between words in the component space.

3.6.5.1 Plotting words into the 3 components space

When the development set of lexical register jokes are plotted into the 3 component space, a score of 36 out of a maximum of 40 is achieved in identifying the incongruous word in a text. The results for detecting lexical register jokes within a set of texts are provided in Table 3.13.

The highest accuracy is 72.5% - a worse result than when the full feature space is used (see Table 3.9). The full feature space was also better at identifying incongruous words - it achieved a

Table 3.13: Precision and recall when computing averages in the 3 component space

joke boundary	accuracy	precision	recall	F score
<= -0.35	29/40 (72.5%)	14/19 (73.7%)	14/20 (70%)	71.8
<= -0.4	29/40 (72.5%)	13/17 (76.5%)	13/20 (65%)	70.3
<= -0.5	25/40 (62.5%)	7/9 (77.8%)	7/20 (35%)	48.3

Table 3.14: Precision and recall when computing averages in the first 2 components space

joke boundary	accuracy	precision	recall	F score
<= -0.4	30/40 (75%)	17/24 (70.8%)	17/20 (85%)	77.3
<= -0.5	28/40 (70%)	13/18 (72.2%)	13/20 (65%)	68.4
<= -0.6	28/40 (70%)	10/12 (83.3%)	10/20 (50%)	62.5

score of 39 out of a maximum of 40. Therefore building a reduced space with the three components suggested by the PCA yielded worse results overall.

3.6.5.2 Plotting words into the first two components space

When we measure the similarity between words using only the first two components of the PCA, the same score - 36 out of 40 - is achieved in identifying the incongruous word in a lexical register joke. The results for detecting lexical register jokes within a set of texts are provided in Table 3.14

Plotting texts into this subspace produces an accuracy of 75%, which is slightly worse than the 80% accuracy achieved when the full feature space is used. Of the 4 cases in which the incongruous word in a lexical register joke failed to be properly identified, 2 of the texts were classified as lexical register jokes. Thus even though the word chosen as the most incongruous was not actually the incongruous word, it was found to be so different from the other words in the text that the text was deemed a lexical register joke. In other words, the algorithm arrived at the right conclusion, but in the wrong way. Given these results, using a reduced space consisting of the first two components of the PCA, does not look promising.

3.6.5.3 1st component space results

In this model, the word with the largest average Euclidean distance from the others on the C1 dimension is the outlier. Euclidean rather than cosine distance is used here because the space has been reduced to a single line (and computing cosine distances between words would only yield a value of either 1 or -1).

Using just component #1, a score of 27 out of a possible 40 results in attempting to identify the incongruous word in the lexical register joke set. Given such a poor result, it was not worth determining how the system performs in identifying lexical register jokes from within a set of texts.

3.6.5.4 2 3 component space results

Using the subspace spanned by the 2nd and 3rd components yielded even poorer results - a score of 11 out of 40 in identifying incongruous words - and it was therefore not worth investigating how the space performs in differentiating between lexical register jokes and other kinds of text. This poor result is surprising because these two components appeared to be adequate estimates of an archaic vs. modern dimension and a ‘common-modern-informal’ vs. ‘common-modern-formal’ dimension. Either this is not the case - the components are inadequate estimates of these

Table 3.15: Precision and recall when computing averages in the components #1 and #3 space

joke boundary	accuracy	precision	recall	F score
<= -0.4	29/40 (72.5%)	15/23 (65.2%)	15/20 (75%)	69.8
<= -0.5	26/40 (65%)	13/20 (65%)	13/20 (65%)	65
<= -0.6	26/40 (65%)	10/14 (71.4%)	10/20 (50%)	58.8
<= -0.7	24/40 (60%)	7/10 (70%)	7/20 (35%)	46.7

dimensions - or they are good estimates but more than just these two dimensions are at work in the development set of lexical register jokes. The latter case would suggest that a subspace comprised of just these two dimensions is too simplistic.

3.6.5.5 1 3 component space results

Plotting the words of a text into the subspace determined by components 1 and 3 yielded a score of 37 out of 40 in identifying the incongruous word in a lexical register joke. The results for detecting lexical register jokes within a set of texts are provided in Table 3.15.

Accuracy is highest at 72.5% when the joke boundary is set at <= -0.4 but this result is (slightly) worse than when words are plotted into the space defined by the first two components of the PCA (see Table 3.14).

3.6.5.6 Conclusion of building a semantic space using PCA components

Building a space with the first two components suggested by the Principal Components Analysis yielded at best an accuracy of 75% in identifying a lexical register joke in a collection of texts. This reduced space therefore performs slightly worse than the full 15 dimensional feature space described in Section 3.5 which yielded an accuracy of 80%. Also, as noted above, finding the most incongruous word in a text using the reduced space yielded a score of 36 out of 40 whereas a score of 39/40 was achieved using the full 15 dimensional feature space. In other words more missteps are made in step #1 of the algorithm (identifying the incongruous word in a text) when the subspace is used, adding further evidence that it should not be preferred over the full 15 dimensional space.

3.6.6 A slight modification to the PCA classification

Section 3.6.5 describes the results of classification testing when using PCA components to build various subspaces. When the words of a text were plotted into a subspace, one of the steps - step #4 in Section 3.6.5 - converted component scores into zscores. We performed this step because each component yields a different range of scores and standardizing these scores (by transforming them into zscores) makes the axes or dimensions of the component space more comparable. Performing this step, however, might be a mistake. It may be that component scores should not be equalized by transforming them into zscores because the extent of the variation in a component's axis may represent something about how important that component is. If this is true then perhaps component scores should be left as they are. We therefore performed once again all the tests described in Section 3.6.5 but this time step #4 was omitted. The results of this testing are described below.

3.6.6.1 Plotting words into the 3 components space

The highest accuracy is 75% - a slightly better result than when zscoring was performed (see Table 3.13) but still worse than when the full feature space is used (see Table 3.9).

Table 3.16: Precision and recall when computing averages in the 3 component space

joke boundary	accuracy	precision	recall	F score
<= 0.6	27/40 (67.5%)	15/23 (65.2%)	15/20 (75%)	69.8
<= 0.55	30/40 (75%)	13/16 (81.25%)	13/20 (65%)	72.2
<= 0.5	27/40 (67.5%)	10/13 (76.9%)	10/20 (50%)	60.6
<= 0.45	27/40 (67.5%)	9/11 (81.8%)	9/20 (45%)	58.1

Table 3.17: Precision and recall when computing averages in the first component space

joke boundary	accuracy	precision	recall	F score
<= 4.5	17/40 (42.5%)	16/35 (45.7%)	16/20 (80%)	58.2
<= 4	16/40 (40%)	15/34 (44.1%)	15/20 (75%)	55.6
<= 3.5	16/40 (40%)	15/34 (44.1%)	15/20 (75%)	55.6
<= 3	14/40 (35%)	12/30 (40%)	12/20 (60%)	48
<= 2.5	15/40 (37.5%)	10/25 (40%)	10/20 (50%)	44.4

3.6.6.2 1st component space results

Using just component #1 as the subspace yielded the poor results shown in Table 3.17. Euclidean rather than cosine distance is used to measure the distance between word vectors because the space has been reduced to a single line (and computing cosine distances between words would only yield a value of either 1 or -1).

3.6.6.3 2 3 component space results

Using components #2 and #3 as the classifier's subspace yielded the classification results shown in Table 3.18.

3.6.6.4 1 3 component space results

Using components #1 and #3 as the classifier's subspace yielded the classification results shown in Table 3.19. This subspace produced only a marginally higher accuracy (82.5% accuracy) than when the full feature space was used (80% accuracy) for classification (see Table 3.9).

3.6.6.5 Conclusion of building a semantic space using PCA components (without zscores)

Using a subspace consisting of the components #1 and #3 yielded only a slightly higher accuracy than when the full 15 dimensional feature space was used. Given only marginal improvement (accuracy improved by only 2.5%), we decided not to prefer it over the full 15 dimensional space.

Table 3.18: Precision and recall when computing averages in the 2 3 component space

joke boundary	accuracy	precision	recall	F score
<= 0	21/40 (52.5%)	20/39 (51.3%)	20/20 (100%)	67.8
<= -0.2	23/40 (57.5%)	19/35 (54.3%)	19/20 (95%)	69.1
<= -0.3	27/40 (67.5%)	19/31 (61.3%)	19/20 (95%)	74.5
<= -0.4	27/40 (67.5%)	16/25 (64%)	16/20 (80%)	71.1
<= -0.5	22/40 (55%)	8/14 (57.1%)	8/20 (40%)	47.1

Table 3.19: Precision and recall when computing averages in the 1 3 component space

joke boundary	accuracy	precision	recall	F score
<= 0.85	30/40 (75%)	17/24 (70.8%)	17/20 (85%)	77.3
<= 0.8	31/40 (77.5%)	16/21 (76.2%)	16/20 (80%)	78.0
<= 0.75	33/40 (82.5%)	16/19 (84.2%)	16/20 (80%)	82.1
<= 0.7	33/40 (82.5%)	16/19 (84.2%)	16/20 (80%)	82.1
<= 0.65	31/40 (77.5%)	13/15 (86.7%)	13/20 (65%)	74.3

3.7 Problem with development set of lexical register jokes

When we first described simple lexical register jokes, we argued, according to our intuition, that they contain a single word whose tone (tone A let's say) is incongruous to the tone of the rest of the text (tone B). This description, which guided our selection of jokes going into the development set, is too vague, however, because it does not specify how tone B is represented in the text. It turns out that tone B can be communicated in a number of ways, making the set of jokes in the development set more diverse and more complicated than we anticipated. For instance we now believe that 15 of the jokes in the development set, communicate tone B lexically (i.e. one or more words, taken individually, possess tone B) but in the five remaining jokes of this set, tone B is conveyed either (a) by a phrase (b) by the context of the passage (c) by important details in the cartoon from which the text was taken:

1. joke #6: "Sticks and stones may break my bones but rhetoric will never hurt me". On closer analysis of the text, the formal tone of the word 'rhetoric' does not appear to conflict with the tone of another word in the text but with the tone of the playground chant as a whole. Our algorithm will fail to notice this opposition because it does not recognize colloquial expressions and their tone. Interestingly, the word 'rhetoric' also clashes with words that are notably absent from the text. The word 'rhetoric' has replaced the words 'names' or 'words' - either word can appear in the popular expression - and because these displaced words are less formal than their usurper, and are invoked by the text in spite of their absence (because the expression, in its traditional form, is so well known), a clash of tone is generated. The classification algorithm only analyses words that actually appear in a text, however, and so clashes of tone such as this, which occur on a paradigmatic level, will not be noticed.

2. joke #9: "When they recommend serving it at room temperature they are referring of course to the rooms of yesteryear". Details of the cartoon from which this text is taken are crucial here: the text is spoken by a man behind the counter at a run of the mill liquor store and he is talking to a woman who is wearing a large winter coat and a frumpy hat. The imagery is important because it creates a prosaic context and the formal word 'yesteryear' opposes this context rather than the tone of any individual word or phrase in the text.

3. joke #12: "Friends we have temporarily lost the video portion of our talk show but will continue to bring you the inane flummery of our panelists". The formal tone of the word 'flummery' does not seem to contradict the tone of another word in this text so much as, perhaps, the context of the text - television talk shows - which are usually informal and banal. Or perhaps the real joke here is that the man on television is being uncharacteristically

honest about the poor quality of the show.

4. joke #13: "Tell me compared to your other victims how would you rate me on sangfroid". Again, the formal word 'sangfroid' does not contradict the tone of any other single word in this passage. The incongruity in this passage is not so much lexical as it is about the man's skewed priorities. He is more concerned about how he comes across to people, even to people who are robbing him, than he is with his personal welfare. His use of the overly formal word 'sangfroid' in such a dangerous context communicates that he is overly concerned with impressing people - even people stereotypically considered to inhabit the bottom end of the social scale - but the formal word 'sangfroid' has no informal counterpart on the lexical level here.

5. joke #16: "Please can I have the thirty cents without the jawboning?" As with the other examples above, analysing joke #16 at the lexical level misses the incongruity in the text. The cartoon from which this text is taken shows a small child asking his father for his weekly allowance. The main opposition here is that a very young child is using such a rude and rather sophisticated (and therefore adult-like) word such as 'jawboning'. The cartoon makes this context clear but our system is blind to this visual cue and cannot infer that a child is speaking here. It can only look for lexical opposition and in this case will find none.

Although the same thing is happening in all the jokes of the development set - a humorous clash of tone is taking place – the multiple ways tone B is represented demonstrate that further subdivisions of the kind of register humour we are interested in, can and probably should be made. Not narrowing our focus will mean having to develop a system that:

- can detect phrases in a text
- has corpora large enough to handle n-grams of various sizes (i.e. phrases) so that frequency counts of phrases in the corpora are not all zero
- can understand the context of a text and infer the tone of that context
- can discern items in the cartoon from which the text was taken and determine what kind of tone they evoke. (For example it would have to detect the details mentioned above in jokes #9 and #16).

These things are clearly beyond the capabilities of the vector space model we have created so far which has been designed to, at best, detect differences in the tone of individual words. Our goal is to construct a model of a particular subclass of humour that is detailed enough to be implemented in a computer program. But if that subclass covers too broad a range of phenomena, our analysis of the humorous mechanisms at work in these kinds of texts will be unnecessarily complicated and the implementation of the model computationally intractable or even impossible.

We will therefore narrow our definition of the simplest kind of lexical joke and say that this is a text in which the tone of a single word (tone A) conflicts with the tone of one or more other words in the text (tone B) and both tone A and tone B are presented lexically. The five jokes in the development set which express tone B in non-lexical ways have therefore been replaced by five

Table 3.20: Scores when computing cosine averages for improved development set

joke boundary	accuracy	precision	recall	F score
<= 0.65	25/40 (62.5%)	20/35 (57.1%)	20/20 (100%)	72.7
<= 0.6	28/40 (70%)	20/32 (62.5%)	20/20 (100%)	76.9
<= 0.55	30/40 (75%)	19/28 (67.9%)	19/20 (95%)	79.2
<= 0.5	**32/40 (80%)**	19/26 (73.1%)	19/20 (95%)	82.6
<= 0.45	31/40 (77.5%)	15/19 (78.9%)	15/20 (75%)	76.9
<= 0.4	28/40 (70%)	12/16 (75%)	12/20 (60%)	66.7

texts randomly selected from our test set of simple lexical register jokes. The five replacement jokes, where the word with tone A is shown in bold and word(s) with tone B are underlined, are:

1. I understand you perfectly. When you say you want to extend your parameters, it means you want **floozies**.

2. Thou shalt not horn in on thy husband's **racket**.

3. Why art thou giving me a hard time? **Eh**? Speak up!

4. Forbearance is the watchword. That triumvirate of **Twinkies** merely overwhelmed my resolve.

5. His grace the lord archbishop has granted me an audience tomorrow. Work up a few **zingers** will you?

The test set of lexical register jokes has therefore been reduced from 22 to 17 items.

3.8 Classification results on improved development set of lexical register jokes

Testing the algorithm described in Section 3.5 (where we compute the average cosine each word forms with the other words in a text and use the lowest of these to decide whether a text is a lexical register joke or not) on the improved development set of lexical jokes, yielded the scores shown in Table 3.20.

The best scores occur when a joke boundary of 0.5 is used: the classifier achieves 80% accuracy, 73.1% precision and 95% recall. These results are identical to the best results achieved when the classification method was tested on the old (and flawed) development set of lexical jokes. (See Section 3.5.2 for details). This is because 4/5 of the faulty texts were classified as lexical register jokes in the earlier test and the same proportion of new jokes (which have replaced the flawed texts) were classified as lexical register jokes in the new test.

Although the five faulty jokes in the old development set were not actually simple lexical register jokes (as explained above), most of them were classified as such because they housed a pathological word (i.e. a word which appears in only a few or no corpora), and this demonstrates a major flaw with our vector space model: pathological words appear as outliers in the space whose average cosine distance to the other words in the text almost always exceeds the joke boundary. In other words, texts containing a pathological word will almost always be classified as lexical

Table 3.21: Scores for cosine averages method on dev set of lexical register jokes and newspaper quotes

joke boundary	accuracy	precision	recall	F score
<= 0.65	34/40 (85%)	20/26 (76.9%)	20/20 (100%)	86.9
<= 0.6	37/40 (92.5%)	20/23 (87%)	20/20 (100%)	93.0
<= 0.55	36/40 (90%)	19/22 (86.4%)	19/20 (95%)	90.5
<= 0.5	38/40 (95%)	**19/20 (95%)**	**19/20 (95%)**	**95**
<= 0.45	35/40 (87.5%)	15/20 (75%)	15/20 (95%)	83.8

register jokes. Chapter 4 describes this problem in more detail and discusses how it might be addressed.

3.9 Adding another set of newspaper texts

All of the lexical register jokes in the development set are bits of dialogue appearing in cartoons, movies and humorous stories whereas the 'regular' text in the development set are sentences randomly selected from a newspaper - none of which is quoted text. That the two sets of texts differ in this respect may represent a flaw in the testing that has been performed up to this point. Ideally we would like to build a set of regular text which differs from lexical register jokes only in terms of the features that make lexical register jokes humorous - in other words to create a collection of negative examples which are "similar in most of their aspects to the positive examples" (Mihalcea and Pulman, 2007). That way, if the classifier performs well in distinguishing between the two sets, we are more confident that it has based its decisions on the presence or absence of the humour-producing feature we are interested in (i.e. incongruity of tone) rather than on differences between the two sets which have nothing to do with humour.

Twenty quoted texts were therefore randomly selected from the November 29 2010 edition of the Canadian Broadcasting Corporation (CBC) website with the hopes of forming a set of 'regular' text which, because it is composed of speech, may be more akin to the style and form of lexical register jokes. Starting at the homepage of the website, articles were randomly selected and searched for quotes. The first quote to appear in an article was selected and truncated after the 17th word because the average number of words in the lexical register jokes set is 15.4 and texts should probably be of comparable size. Only one quote per article was selected in an attempt to build a varied sample of texts and to avoid bias that might be introduced by taking all the examples from only a few articles. (If all the quotes were from an article on quantum physics, for example, these might be full of esoteric and domain-specific jargon which would make the sample less representative). In this way newspaper quotes from 20 different articles, all of which are listed in Appendix B, were collected.

When the average cosine method (introduced in Section 3.5) is used to distinguish lexical register jokes in the development set from newspaper quotes, the scores listed in Table 3.21 occurred. The highest accuracy achieved is 95% when the joke boundary is set to 0.5.

The excellent results of the average cosine method are encouraging but it would be prudent not to overestimate them because:

1. It may be that the newspaper quotes, although randomly selected, are atypically bland examples of quoted text. Take the following text, for example: "We need to work with the family

in making sure they find the objects, and we need". Only five words, fewer than 1/3 of the words in the text, are content words (i.e. not stopwords) and the vocabulary choice seems particularly simple and uniform. In general, speech is probably simpler and less varied than written text, but perhaps not this simple!

2. We are not really comparing humorous quoted text with 'regular' quoted text after all. Lexical register jokes in the development set are not actually examples of speech - they are creative works of fiction which are presented as spoken text. It is possible that the vector space could be classifying texts based on lexical differences between actual spoken text and creatively written text which is only masquerading as spoken text.

3.10 Summary

Lexical register jokes are texts which owe their humour to the presence of a clash of tone. The simplest kind of lexical register joke is one in which the tone of a single word is incongruous relative to the tone of the rest of the text and we are interested in building a semantic space which can provide an objective and quantifiable way of measuring this kind of humorous incongruity. After experimenting with different distance metrics (Euclidean, Mahalanobis, cosine), different kinds of pre-processing (tf-idf, log entropy pre-processing) and various sets of corpora, the best solution we found involved:

- computing frequency counts of words in the 15 corpora of corpus set D (see Appendix A for a list of the corpora in this set). These corpora display varying degrees of lexical formality and represent the axes or dimensions of a semantic space that will be used for measuring differences in tone.
- normalizing and performing log entropy transformations of the frequency counts.
- using the cosine metric to compute similarities between word vectors.

Using this semantic space, a score of 39 out of a possible 40 was achieved in identifying the incongruous words in a development set of simple lexical register jokes. Experiments were then performed to determine if the space could be used to automatically detect these same lexical register jokes from within a collection of texts. The best accuracy achieved was 80% (with a precision of 73.1% and a recall of 95%), and attempts were then made to improve these results using Principal Components Analysis (PCA).

Building a reduced space consisting of the first 2 components suggested by the PCA, lowered the accuracy of detecting lexical register jokes to 75%. This drop in accuracy can partially be accounted for by the fact that when the subspace is used, more errors are made in the initial steps of the algorithm in finding the incongruous word in a text.

A problem with five jokes in the development set was then noticed: in each of these texts, one of the conflicting tones was represented non-lexically. In order to maintain our goal of constructing a model of a subclass of humour that can be implemented computationally, we therefore made our description of the simplest kind of lexical register joke more precise. The simplest lexical register jokes contain a single word whose tone conflicts with the tone of other words in the text and both tones are expressed lexically (not phrasally or contextually, for example). The five faulty jokes in

the development set were therefore replaced and the average cosine method was performed on the new development set of jokes. Results were very similar to those yielded by tests on the older (and faulty) set of development texts and this revealed a fundamental problem with the vector space model we have developed so far: texts which contain a pathological word will almost always be classified as lexical register jokes. This problem will be addressed in the next chapter.

It was also noticed that the simple lexical register jokes in the development set are all examples of quoted speech whereas the set of 'regular' texts used in testing so far has consisted of written text from newspapers. An additional set of 'regular' text, consisting of quotes from newspapers, was therefore assembled and used to test the average cosine method. The average cosine method yielded a particularly high 95% accuracy in distinguishing between simple lexical register jokes and the new set of regular text. These results are encouraging but should be regarded with some skepticism because, among other things, the new set of regular text may be particularly bland.

The next chapter discusses the difficulty certain types of words pose for the vector space model, and in Chapter 5 we will explore using clustering solutions to automatically distinguish between lexical register jokes and regular text. These algorithms produce results on par with the best methods described in this chapter, and they also offer the possibility of detecting more complicated lexical register jokes, rather than just the simplest kind.

Chapter 4

Data Sparsity

4.1 Introduction

The previous chapter introduced a classifier which aims to distinguish lexical register jokes from regular text. The classifier makes use of a vector space into which individual words of a text are plotted. The aim is to have a word's position in the space reflect its tone, and so large distances between words would therefore represent significant differences in their tone. Significant difference of tone is a fundamental characteristic of lexical register jokes and if the vector space accurately reflects these differences, the classifier will be able to recognize these kinds of texts.

Various parameters of the classifier were experimented with and a development set of lexical register jokes and 'regular' text was used to evaluate the classifier's performance. The best accuracy achieved on the development set was 80% with a precision of 73.1% and a recall of 95%.

A step in the classifier's algorithm involves computing frequency counts of a word in corpora that vary in tone and these frequencies represent the word's coordinates in the vector space. One of the assumptions of our model is that frequency count distributions in the corpora can act as a kind of "tonal fingerprint" of a word. If the corpora are too sparse, however, these distributions will be compromised and may fail to be proper estimates of tone. The aim of this chapter is to determine the extent to which corpus sparsity might be responsible for classifier errors and to explore what might be done to correct some of these errors.

4.2 Frequency counts as estimators of tone

How do we regard a word that has a low frequency count in a given corpus, such as the King James version of the bible? Can we infer anything about its tone? Words like 'Amminadib', 'cheereth' and 'clappeth', for example, occur only once in the bible and they possess a formal tone. On the other hand, the words 'folks', 'onions' and 'moist' also appear only once in the bible yet these words are not archaic or formal sounding. From this perspective, the task of evaluating the tone of words using frequency counts in a corpus seems daunting.

The task looks even more formidable when we consider that the number of words occurring just once or twice in any given corpus is often considerable. For example there are 12,584 word types in the version of the King James bible we are using and nearly a quarter (3962) of these appear in the text only once. Nor is the bible unique in this regard. Word types "have a very uneven distribution" and "very rare words make up a considerable proportion" of most texts (Manning and Schutze, 1999). Manning and Shutze write "it is hard to predict much about the behaviour of words that you never or barely ever observed in your corpus" and the same could be said about predicting

the tone of infrequently occurring words.

It is important to keep in mind however, that we are not looking at a word's frequency count in a single corpus, in order to estimate its tone, but at its frequencies in multiple corpora. We find, for example, that the word 'Amminadib' does not appear in any of the other corpora in corpus set D, for example, and this makes its single appearance in the bible – a seemingly tenuous connection to this corpus – more significant when regarded in this light. This is because words which are rare in a collection of corpora are probably more representative of the corpora to which they belong. Whereas the word 'folks' occurs only once in the bible but appears in 11/15 of the corpora in corpus set D, making its tenuous connection to the bible corpus (and perhaps its tone) even more tenuous.

Because their profiles in the various corpora are so different, these two words, which seemed alike when their frequencies in the bible alone were considered, will be far apart from each other in the vector space of our model. In this case, the vector space model properly shows that these two words have noticeably different tone. (Whether the vector space would classify a text housing these two words as a lexical register jokes is another matter).

But sometimes estimating difference in tone using the vector space model fails and one reason for this might be on account of corpus sparsity (making the frequency count vectors inaccurate in some way and the distances between them misleading). This chapter attempts to investigate the extent to which sparsity of corpus data accounts for errors made by the classifier.

4.3 Words that are problematic for the classifier

Numerous words in the development set of jokes appear in only a few corpora or none at all. In some cases this is appropriate. Words like 'logy' and 'triumvirate' are truly rare and their esoteric nature is used to generate incongruity in a text when they appear side by side with, respectively, the relatively informal words 'mom' and 'twinkies'.

On the other hand, other words in the development set such as 'yum' and 'mommy' (which appear in jokes #3 and #12 respectively) have low frequencies in the subset of corpora in which they appear, probably not because they are truly rare but because the set of corpus data we are using (corpus set D) is simply too sparse.

Sparsity of corpus data, a perennial issue in NLP, is perhaps the biggest problem with our vector space model. To better understand this problem, we have created a partial taxonomy of the different types of frequency count distributions that can occur in the space (see Table 4.1). The left column of this table lists the profiles (A - C) that are especially difficult to interpret and we will examine each of these in turn.

4.3.1 Profile A

Words with profile A are problematic, not because they appear in just a few corpora but because their frequencies in those few corpora are low. The phenomenon of words appearing in just a few corpora is to be expected. For example archaic words such as 'thee' and 'thou' will, for the most part, appear in only a subset of the corpora (e.g. archaic corpora) and these common ancient words will have high frequencies. Words with high frequencies in just a few corpora have profiles D or E in Table 4.1 and are not considered as problematic as words with profiles A - C. This is because high frequencies in just a few corpora and zero or low frequencies in the remaining corpora gives

Table 4.1: Different frequency count distributions

Problematic Profiles	Non-problematic Profiles
A low frequencies in just a few corpora (and zeroes elsewhere)	**D** high frequencies in just a few corpora (and zeroes elsewhere)
B low frequencies in many corpora (and zeroes elsewhere)	**E** high frequencies in just a few corpora (and low frequencies elsewhere)
C words with zero vector profiles (i.e. do not appear in any corpora)	**F** high frequencies in numerous corpora

us higher confidence that a word with such a profile is closely associated with the corpora in which it appears (and perhaps conveys some of the tone of those corpora).

Words with profile A are more difficult to interpret, however, because words with differing tone can belong to this set. For example, a word with profile A could be:

1. a word that rarely occurs in the subset of corpora but those corpora still reflect the tone of the word. Let's say the word 'electrodynamics' appears in a subset of scientific corpora but nowhere else. These corpora do convey the tone of the word - it sounds scientific - but it occurs only rarely in these corpora, perhaps because the corpus data is sparse (and so this word should actually have, for example a profile of D or E). Or it may be the case that low frequencies are appropriate, because 'electrodynamics' is a highly technical term and is rarely used outside very specific contexts, even in scientific journals.

2. a word that rarely occurs in the subset of corpora and these corpora do not reflect the tone of the word. For example let's say that the subset of corpora in which the word occurs is once again a set of scientific texts. A word such as 'squeezing' might occur only rarely in these corpora and it does not sound particularly scientific.

3. a word that is rarely used anywhere (e.g. 'triumvirate'). The dominant tone of these rare words might be that they sound rare. Hence their tone may not be conveyed by the few corpora in which they appear. In our example, the word 'triumvirate' sounds formal and rare in a literary way rather than a scientific way and its tone is not accurately conveyed by the subset of corpora in which it happens to appear.

Thus from the examples above, we see that a word with profile A which occurs infrequently in a subset of corpora might communicate the tone of those corpora (e.g. 'electrodynamics' in the example above), might have a neutral tone ('squeezing') or might sound predominantly esoteric ('triumvirate').

In fact distinguishing between the different kinds of words that can have profile A is complicated further because the classifier uses the cosine metric to measure the similarity between

vectors. Under the cosine method, a vector of [1,1,0,0,0], for example, is identical to the vector [100,100,0,0,0] because the cosine method normalizes vectors. After normalization, these two vectors will be of unit length and point in the same direction. Thus, in the vector space used by the classifier, a word with profile A is not only ambiguous for the reasons listed above, but it is also indistinguishable from some words with profile D (see Table 4.1).

The reverse is also true of course. Normally profile D would be one of the least ambiguous profiles - a relatively clear indicator that a word truly belongs to a subset of corpora and has a distinctive kind of tone - yet under the cosine method, words with profile D are conflated with a set of words that have a profile which is very difficult to interpret (profile A). In other words, one of the few profiles we considered relatively non-problematic has conjoined with a profile that is perhaps the most difficult to interpret!

4.3.2 Profile B

Words with profile B are also difficult to interpret. For instance it may be the case that a word with profile B would more accurately exhibit profiles D or E had the corpora data been fuller[1]. Words with profile B will probably not appear as outliers in the space because, under the cosine method, they will appear close to common words which have high frequencies in numerous corpora (profile F). (For example a word with vector [1,2,1,1] will be identical to a word with a vector such as [100,200,100,100] when the cosine method is used to measure similarities in the vector space. This is, as explained above, due to the normalization of vectors that occurs when computing cosine distances). Thus a word which misleadingly exhibits profile B because of corpus sparsity, might actually be a word with a more distinctive tone than this profile suggests and contributing to an incongruity of tone in the text. It will not stand out in the vector space, however, and might be indistinguishable from a crowd of common words clustered near it.

4.3.3 Profile C

A word with profile C is a zero vector. Recall that the cosine distance between a zero vector and another vector is undefined. (The cosine of two vectors u, v is defined as their dot product divided by the product of their lengths. When u or v is a zero vector, the result is 0/0). When faced with having to compute the cosine distance between a zero vector and another vector, we have elected to have the classifier output a cosine distance of 0. A cosine of 0 (which represents a 90° angle) is the maximum distance two vectors can be from each other in the space. (Vectors in the space can not be at an angle greater than 90° because they do not contain negative values. A word cannot appear a negative number of times in a corpus). Consequently any text containing a word with profile C will exceed the joke boundary and be considered a lexical register joke.

We decided to have the classifier behave in this way because we believe that zero vectors are not always mistakes arising from corpus sparsity. lexical register jokes sometimes make use of truly rare words to create opposition of tone. In fact, of the twenty jokes in the development set, we believe that 3/20 (15%) of these texts use very rare words for humorous effect[2]. The appearance

[1]For that matter, it could be the case that a word with almost any profile 'should' have any one of the other profiles, depending on the extent and kind of corpus sparsity impairing a system.

[2]The three jokes which we believe use rare words (shown in bold below) to create opposition of tone are:

1. Mom! Bart is on a strict diet of complex carbohydrates. Steak will make him **logy**.
2. Last chance to see the beautiful reflections in Mirror Lake before **eutrophication** sets in.

of zero (or nearly zero) vectors in the vector space is therefore appropriate sometimes and should not always be dismissed as noise. Such a vector sometimes communicates valuable information about a word (i.e. that it is rare) and reflects an important characteristic of a certain type of lexical register joke.

The problem of course is distinguishing between the case when a word with profile C actually is rare and when it only appears to be, because of corpus sparsity. Even if a word with profile C is actually rare, it is naive to have the classifier regard every text housing such a word as a lexical register joke because, obviously, non-humorous texts will also contain such words. Our decision to have the classifier regard a text containing a profile C word as a lexical register joke might not be entirely misguided if such words appear more frequently in lexical register jokes than in 'regular' text, but whether this is in fact the case is not clear. Classifiers conventionally must classify all of the data they are presented with, however, and in the face of this requirement, some kind of decision, however arbitrary and unsatisfying it appears to be, had to be made with regards to texts containing these words. The joke generator, however, is not constrained in this way and so, given the ambiguities of profile C words, texts containing them will be eliminated as candidate jokes. See Chapter 7 for details.

4.3.4 Non-problematic profiles D, E, and F

Profiles D and E provide perhaps clearer evidence that words possess the distinctive tones of the corpora in which they appear. High frequencies in just a few corpora and zero (profile D) or low (profile E) frequencies in the remaining corpora gives us higher confidence that a word with such a profile is closely associated with the corpora in which it appears (and perhaps conveys some of the tone of those corpora).

Profile F is probably the clearest indicator of a word's tone (or lack thereof) because it is likely that only stopwords and very common words will have high frequencies in numerous corpora. Thus words with profile F are likely to have a neutral tone.

4.4 Distinguishing between profile A and D words

As stated above, predicting the tone of profile D words would normally be relatively straightforward but when the cosine metric is used to measure the similarity between words in the vector space, profile D words are conflated with profile A words - words whose tone is very difficult to predict. A way of distinguishing between words with these profiles, however, would be to make use of a threshold. For example a word belonging to profile A would be a word that has frequencies in only a few corpora (and zeroes elsewhere) and each of those frequencies are below the threshold value, whereas a word with profile D also appears in only a few corpora but its frequencies are above the threshold. The classifier could then be equipped with confidence ratings. For each text it encounters, the classifier could generate a number representing its confidence in classifying that text, and texts containing profile A words would receive lower confidence ratings than other kinds of texts.

A practical problem emerges however with the introduction of confidence ratings: how would

3. Forbearance is the watchword. That **triumvirate** of Twinkies merely overwhelmed my resolve.

the classifier's performance be evaluated under such a system? Would accuracy scores be computed only on texts for which the classifier had a confidence level higher than 80% for instance? How many texts in the development or test sets would actually be classified if this were the case? Given the rarity of lexical register jokes, filtering out texts with low confidence ratings may reduce the size of the development and test sets by too much.

Perhaps because of difficulties such as these, classifiers traditionally do not abstain from classifying certain data and must classify all of their input. We will therefore follow this convention and will not incorporate confidence levels into the classifier. A confidence rating system will be implemented in the lexical register joke generator, however. When the vector space is used to generate lexical register jokes, the generator will be directed to output only texts about which it is fairly confident. If a text constructed by the generator is found to contain a word with profile A, for instance, that text will be discarded because estimating whether it contains an incongruity of tone is too uncertain. Chapter 7 discusses this and other kinds of filtering performed by the joke generator.

4.5 Testing whether words with profile A or D always appear as outliers

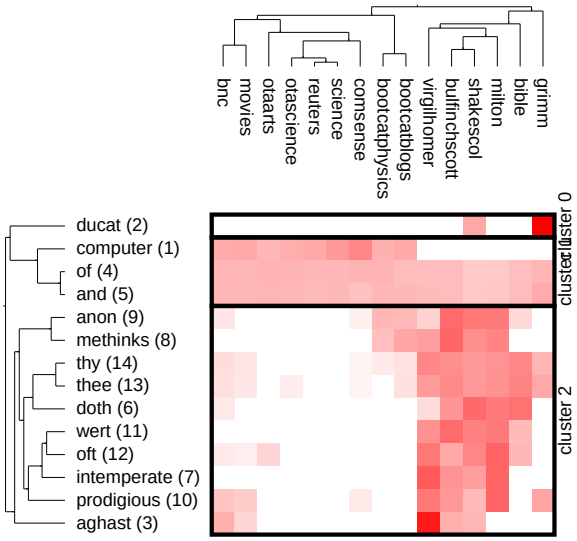
We know that words with profile C will always appear as outliers in the space because, rightly or wrongly, we chose to have the classifier treat them this way. But what about words which occur in only a few corpora? Does the classifier automatically regard a word with profile A or D as the most outlying word, regardless of the frequency count distributions of the other words in the text?

Sometimes, of course, a word with profile A or D will be one of the incongruous words in a text, but we want to know whether it always appears as an outlier (i.e even when it is not responsible for incongruity of tone in a text). This is a real possibility when we recall that stopwords and common words in texts are plotted into the vector space along with content words. These common types of words are included because early tests of the classifier determined that doing so made accuracy scores significantly higher. Stopwords and common words have high frequencies in all the corpora and so their vectors will tend to point in roughly the same direction. A single text will often contain numerous stopwords and common words and consequently words with profile A or D might always stand out as outliers in such contexts.

This is an important issue to examine. If we arbitrarily define "a few corpora" to be three or fewer corpora (but not 0 corpora) and words with "low frequencies" as those that appear no more than 8 times (per million words) in a corpus, then 8/20 of the lexical register jokes in the development set contain a word with profile A and 1/20 contains a word with profile D. The classifier regards each of these words as the most outlying word in its text and all 9 of the texts to which these words belong were properly classified as lexical register jokes. But the worry here is that these texts (9/20 or 45% of the jokes in the development set) may have been classified correctly, not because an incongruity of tone was detected but simply because each of these texts contains a word that occurs in only a few corpora (and distances to these kinds of words might always exceed the joke boundary).

Further doubts arise about whether the classifier properly handles profile A and D words when we find that four of the newspaper texts contain words with profiles A (3/20) or D (1/20) -

Figure 4.1: Bag of words test to determine if rarest word is always the most outlying word



each of which is the most outlying word in the text - and all four of these texts were consequently classified as lexical register jokes. (14 newspaper texts which did not contain these types of words were correctly classified and 1 newspaper text which contained a zero vector was classified as a lexical register joke).

In an attempt to see whether the classifier automatically regards a word with profile A or D as the most outlying word, regardless of the frequency count distributions of the other words in the text, the following "bag of words" were intuitively and informally selected and were plotted into the vector space:

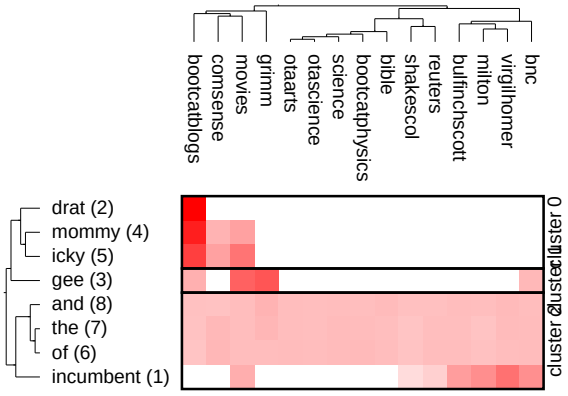
> wert thee thy methinks doth and of the intemperate prodigious ducat computer oft aghast anon

(A "bag of words" can be used here because the text does not have to be grammatically or semantically correct in order investigate this matter). A handful of stopwords appear in this constructed text because stopwords regularly appear in lexical register jokes and, along with content words, are plotted into the vector space for the consideration of the classifier.

The word in our bag of words test which occurs in the fewest corpora is the word 'ducat': it appears in only two corpora with frequencies of 7.1 and 1.82 times per million words and so this is a word with profile A. This word is archaic and is in keeping with the tone of most of the other words in the text which are also archaic and formal sounding. The word 'computer', however, also appears in the list and we believe that it is the most incongruous word in the list. It appears fairly frequently - in all but six of the corpora - and we are interested in seeing whether the classifier will regard it or the word 'ducat' as the most outlying word in the space. Figure 4.1 shows the frequency count distributions for these words.

When this text is processed by the classifier, the word 'computer' is in fact regarded as the most outlying word in the vector space when the average cosine method is used. (In Figure 4.1, the number 1 appears in brackets after the word 'computer', indicating that its distance from the other words in the text is the greatest).

Figure 4.2: Bag of words test 2



In a similar test, the words "gee mommy incumbent and the of icky drat" are plotted into the space. Of all the words, the word 'drat' appears in the fewest number of corpora - only one, the corpus of blogs - and it appears there only 2.54 times per million words. It too is a word with profile A and its tone is in keeping with the other content words of the text except for the formal word 'incumbent', which we believe is the most incongruous word in the text. Figure 4.2 shows the frequency count distributions for this second bag of words test and demonstrates that once again, the classifier did not simply choose the rarest occuring word as the most outlying word: the word 'incumbent' is properly identified as the outlier in the space even though it appears in significantly more corpora (6 in total) than the word 'drat'. These results are encouraging because they demonstrate that a word with profile A does not always appear as the most outyling word in the vector space.

This also means that words with profile D will not always appears as the most outlying word in a vector space. As explained earlier, the cosine method normalizes vectors. Thus a word vector with profile A (e.g. [1,1,0,0...] and a word vector with profile D [300,300,0,0...] - both of which are pointing in the same direction - are both normalized to unit length, and so become identical. As our two bag of words tests have demonstrated, words with profile A do not always appear as the most outlying word in a text. If we were to replace the words with profile A in the bag of words tests with profile D words which point in the same direction, then, after normalization, these profile D words will be identical to the profile A words. They too will therefore not always appear as the most outlying words in the vector space.

4.6 Improving corpus coverage

Twelve lexical register jokes in the development set contain a word which rarely occurs in any of the corpora or not at all (i.e. words with profile A or C) but we believe that at most only three of these words are actually rare: 'logy', 'eutrophication' and the word 'triumvirate'. That leaves 9 texts containing words which may not actually be rare and occur infrequently only because corpus data is sparse. This means that nearly half of the lexical register jokes in the development set have been correctly classified as lexical register jokes, but perhaps for the wrong reason.

In light of this, we decided to introduce a new and larger set of corpora (corpus set E). Informal corpora such as lyrics from rap music (~2.2 million words), and the first 11 seasons of the television show South Park (~800,000 words) were added to corpus set D, as were formal corpora

such as book reviews (~3.4 million words) and science articles (~2.1 million words) written for the New York Times newspaper. Corpus set E thus contains 20 million words - roughly twice as much data as corpora set D.

It was hoped that adding this data would help reduce the number of misclassifications made on account of corpus sparsity and might also provide fuller coverage of the following kinds of tone (as informally assessed):

1. archaic
2. modern formal (scientific)
3. modern formal (literary)
4. modern informal
5. middle of the road formality

These are only informal and intuitive categories and are not used in any way in the formal model. A full listing of corpus set E, organized in terms of these categories, appears in Appendix A. We have chosen to represent these categories in particular because we believe that the following oppositions of tone are prevalent in lexical register jokes:

- 1 vs. 4
- 2 vs. 4
- 3 vs. 4
- 1 vs. 2

Sparsity of corpus data will always be a problem because any set of corpora chosen will be limited, but the addition of more data will hopefully reduce the magnitude of the problem.

4.7 Testing of corpus set E

When the classifier uses corpus set E to create a vector space and the average cosine method (which yielded the best classification results in the testing described in chapter 3) is used to estimate whether a text is a lexical register joke or not, an accuracy score of 31/40 (77.5%), a precision of 19/27 (70.4%) and a recall of 19/20 (95%) result. (see Table 4.2)[3].

Although using corpus set E did not improve test results (when corpora set D was used, the classifier's accuracy was 80%, precision was 73.1% and recall 95%), it may nevertheless be worthwhile using this new set of corpora. For instance 12 of the lexical register jokes in the development set contained words with profiles A or C[4] when frequencies of words were computed in corpus set D and that number has been reduced to 8 texts when corpus set E is used to create profiles of words[5]. This might mean that more of the lexical register jokes are being classified

[3]Also, when newspaper quotes rather than newspaper text are used as non-humorous texts, results are identical to when corpus set D was used: an accuracy score of 38/40 (95%), a precision of 19/20 (95%) and a recall of 19/20 (95%) result.

[4]Using corpus set D, 8 lexical register jokes contained a word with profile A and 4 contained a word with C.

[5]Using corpus set E, 8 lexical register jokes have a word with profile A and no lexical register jokes have a word with profile C.

Table 4.2: Results of using average cosine method with corpus set E

joke boundary	accuracy	precision	recall	F score
<= 0.65	29/40 (72.5%)	20/31 (64.5%)	20/20 (100%)	78.4
<= 0.6	**31/40 (77.5%)**	19/27 (70.4%)	19/20 (95%)	80.9
<= 0.55	29/40 (72.5%)	16/23 (69.6%)	16/20 (80%)	74.4

correctly, not because they possess words which falsely appear to be rare but because an opposition of tone was detected. If this is the case then the new set of corpus data may prove more useful when testing unseen data and when it comes time to generate new lexical register jokes.

4.8 Other classifier errors involving problematic words

The highest accuracy score results when the joke boundary is set to 0.6. At this boundary, the classifier misclassifies only one lexical register joke but 8 newspaper texts – nearly half of the development set of 'regular' text. In other words, the classifier is only marginally better than random when classifying the regular text of the development set.

Only one of the newspaper texts of the development set contains a word with a problematic profile when corpus set E is used (4 newspaper texts contained a profile A or C word using corpus set D)[6]. (Newspaper text #19 contains the word 'contorts' which is a profile A word). The outlying word in the remaining seven texts appear in at least 5 corpora: 'ebbing' appears in 5 corpora, 'tallies' occurs in 6, 'rite' in 12, 'municipalities' in 8, 'shrieked' in 6, 'mobbed' in 5, and 'rite' (again) in 12. Perhaps not surprisingly, all of these outliers except for one are also the rarest word in the text, and it seems likely that in spite of having doubled the size of the corpus data, sparsity of data is still to blame for so many misclassifications of 'regular' text. We cannot add more corpus data indefinitely, however, and so attempts were made to address the problem in new ways by

1. using Laplace smoothing, a pre-processing method.

2. implementing a new Pointwise Mutual Information (PMI) algorithm which uses corpus set E to identify which word is most incongruous with the other words in a text but then goes elsewhere - to Altavista and the vast resources of the world wide web - to estimate the extent to which this word is relatively incongruous.

4.9 Smoothing

Chen and Goodman write that "whenever data sparsity is an issue, smoothing can help performance" (1998). The name smoothing is used because frequency count distributions are 'smoothed' or made "a little less jagged" by subtracting counts from ngrams with high frequencies and redistributing those counts to ngrams with low or zero frequencies. Smoothing techniques are often used to construct n-gram language models and so many of these algorithms deal with ngrams where n > 1. One of the simplest kinds of smoothing, Laplace smoothing (also sometimes called

[6]Our definitions of profile A and D should probably be revised, however, if corpus set E is adopted. When corpus set D was used, which consisted of 15 texts, these were words which appear in three or fewer corpora. corpus set E is larger - it consists of 25 texts - so perhaps profile A or D words should be redefined as words that appears in five or fewer corpora. If this is the case, the number of newspaper texts containing a profile A word when corpus set E is used is 2/20 rather than 1/20.

Table 4.3: Laplace smoothing, using avg cosine method

joke boundary	accuracy	precision	recall	F score
<= 0.88	28/40 (70%)	19/31 (61.3%)	19/20 (95%)	74.5
<= 0.85	32/40 (80%)	15/18 (83.3%)	15/20 (75%)	78.9
<= 0.80	27/40 (67.5%)	7/7 (100%)	7/20 (35%)	51.9

Add-1 smoothing), however, is applicable to unigrams (n = 1) and might therefore be useful in our context in which the frequencies of individual words are being computed.

4.9.1 Laplace smoothing

Without smoothing, the probability P of a word w_i appearing in a particular corpus is estimated as $P(w_i) = \frac{c_i}{N}$ where c_i is the word's frequency count in the corpus and N is the total number of word tokens in the corpus. The idea behind Laplace smoothing is that we pretend to see every word once more than we actually did. "Since there are V words in the vocabulary and each one was incremented, we ... need to adjust the denominator to take into account the extra V observations" (Jurafsky et al., 2009) and so the smoothed probability of a word becomes $P_{laplace}(w_i) = \frac{c_i+1}{N+V}$.

Take for example the word 'the' which occurs 4600 times in the bible corpus. Before smoothing, its probability of appearing in this corpus is 4600/816914 which is about .56%. After smoothing, this probability is only slightly reduced to 4601/(816914 + 12584), roughly 0.55% and this seems reasonable because we do not want to greatly reduce the probability of words which occur quite frequently in a corpus. We do not want to do so because if a word has been seen frequently in a corpus, its unsmoothed probability may be quite reliable and should therefore not be significantly altered.

Conservative reduction and redistribution of probability mass is probably appropriate in the case of common words but such conservatism appears to be an ineffective way of dealing with problematic words (words with profiles A - C for instance). As we have seen, the word 'Amminadib' appears only once in the bible corpus and so its probability before smoothing is 1/816914 or $1.22 * 10^{-6}$. Laplace smoothing increases its probability of appearing to 2/(816914 + 12584) which is $2.41 * 10^{-6}$. After smoothing, the probability of this word is twice as high but is still practically zero and it is doubtful that such slight adjustments to problematic words will produce significant improvements in the classifier's performance.

These doubts are borne out in tests of the classifier. When Laplace smoothing is incorporated into the system and the average cosine method is used to automatically distinguish the lexical register jokes and regular texts of the development set, accuracy improves only slightly: from 77.5% to 80% accuracy (32/40). See Table 4.3 for details.

Also, we find that when smoothing is performed, the pair of words that are furthest apart in the classifier's vector space are less intuitive than when smoothing is not performed. Before testing, we marked by hand the pair of words in lexical register jokes which we felt were most incongruous. When Laplace smoothing is not performed, 14/20 of the pairs of words deemed most incongruous by the classifier match our intuitions whereas only 8/20 pairs match our intuitions when smoothing is performed. See Table 4.4. (Note that the "manually labelled" column sometimes contains more than one pair of words. Recall from Chapter 3 that we define simple lexical register jokes as texts

in which the tone of a certain word (tone A) conflicts with the tone of one or more other words (tone B). In some cases, therefore, a word with tone A will conflict with multiple words that have tone B. Thus when numerous word pairs occur in this column, this means that we believe that a conflict of tone is expressed by any one of these pairs. And we were interested to see whether the classifier, when it finds the most incongruous pair of words in a text, chose any of the pairs in a given list).

Given only marginal improvement in accuracy and much less intuitive selection of incongruous words when Laplace smoothing is performed, we decided against integrating this kind of smoothing into the system. Lacking credible frequencies for a word cannot be remedied simply by adding 1 (or some other constant) to each frequency. The problem with low frequency counts resulting from corpus sparsity is that the frequencies of problematic words are probably accurate in some dimensions but significantly wrong in others and such a coarse-grained method as Laplace smoothing is unable to correct these kinds of errors.

4.9.2 Simple Good Turing smoothing

Another smoothing algorithm - Simple Good Turing - was also investigated as a way of dealing with problematic words but proved to be unhelpful because it does not apportion probabilities to individual words occurring 0 times in a corpus. Instead it estimates a total probability of all unseen objects (Gale and Sampson, 1995). For example Simple Good Turing might estimate that the total probability for unseen words in a given corpus is 0.2048. If the vocabulary size of that corpus is V, then the number of unseen words in the corpus is V subtracted from the vocabulary size of English as a whole. But what is the vocabulary size of English as a whole? Even if we were able to make this calculation, we would then have to ask whether the probability mass of 0.2048 should be distributed evenly among all unseen types. Considering the difficulty of these questions - questions which would have to be answered in order to make use of the Simple Good Turing algorithm - this smoothing technique was abandoned as a possible way of dealing with problematic words.

4.10 Using PMI to improve detection of lexical register jokes

Another way we hoped to address the corpus sparsity problem was to implement a new algorithm which uses the web as a giant corpus to estimate the co-occurrence of words. Information about how strongly words co-occur is used, in a variation of the Pointwise Mutual Information (PMI) method, to estimate how incongruous a word is compared to other words in a text.

The main assumption of the PMI method, described in Turney and Littman (2003), is that words which score similarly on certain dimensions tend to cluster together in texts. In the paper, the authors are interested in automatically determining the semantic 'orientation' of a word - whether a word 'sounds' positive or negative. (For example the words 'cries' and 'disturbing' are negative while 'honest' and 'intrepid' sound positive). The authors created by hand the following list of seven positive words and seven negative words which were used as paradigms of positive and negative semantic orientation:

Pwords = {good, nice, excellent, positive, fortunate, correct and superior}

Nwords = {bad, nasty, poor, negative, unfortunate, wrong, and inferior}

Table 4.4: pairs of words with lowest cosine when smoothing and not smoothing

joke no.	no smoothing	smoothing	manually labelled
1	operator personage	operator personage	operator personage
2	mom logy	mom strict	mom logy
3	as yum	chef he	chef yum
4	appointments rapping	appointments rapping	appointments rapping
5	maximization moola	gentlemen management	maximization moola, gentlemen moola, management moola, profit moola
6	diagnosis tuckered	data tuckered	diagnosis tuckered, data tuckered
7	development messing	research man	research messing, development messing
8	wield tart	executive sword	executive tart, supreme tart
9	fellas crestfallen	are gonna	fellas crestfallen, gee crestfallen, superbowl crestfallen,
10	gee determining	gee determining	gee clients, gee simultaneously, gee potential, gee yield
11	diminutive do	big diminutive	big diminutive
12	demands mommy	demands mommy	damn mommy
13	customer merriment	customer merriment	customer merriment
14	beautiful eutrophication	eutrophication sets	beautiful eutrophication
15	daddy midmorning	daddy profit	daddy market, daddy account, daddy profit, daddy midmorning
16	parameters floozies	you parameters	parameters floozies
17	shalt racket	shalt racket	shalt racket, thou racket, thy racket
18	thou eh	thou eh	thou eh
19	forebearance watchword	forebearance merely	triumvirate twinkies, forebearance twinkies, watchword twinkies, overwhelmed twinkies, resolve twinkies
20	lord zingers	archbishop tomorrow	archbishop zingers, lord zingers, grace zingers

Their system then attempts to infer the direction and intensity of the semantic orientation of a word from its statistical association with these paradigm words. The strength of a word's association with each of the paradigm words is estimated using the following:

$$PMI(word1, word2) = \log_2\left(\frac{p(word1 \& word2)}{p(word1) * p(word2)}\right) \backsimeq \log_2\left(\frac{\frac{1}{N}hits(word1 NEAR word2)}{\frac{1}{N}hits(word1) * \frac{1}{N}hits(word2)}\right)$$

(Note that N is the total number of documents indexed by Altavista). For a given word, queries are issued to Altavista and the number of hits in which a word appears with each of the paradigm words is computed. The number of times a word occurs with another word can only be approximated - Altavista does not provide the exact count - by counting instead the number of hits (i.e. matching documents) in which the two words co-occur. Altavista was chosen because it has a NEAR operator which will only return documents that contain the words within 10 words of each other. The overall semantic orientation of a word is then computed by:

$$SO - PMI(word) = \sum_{pword \in Pwords} PMI(word, pword) - \sum_{nword \in Nwords} PMI(word, nword)$$

One way of using the PMI method to automatically detect lexical register jokes would be to perform the $SO - PMI$ computation for each word in a given text, but with paradigm words which exemplify formality and informality rather than semantic orientation. Formal and informal paradigm words could be selected by hand and estimates of a word's formality or informality could then be made by measuring how often it co-occurs with these exemplars. Words in a text could be scored in this manner and if some words receive significantly different PMI scores from their counterparts, one might conclude that the text is a lexical register joke.

We adopted a different approach however - one which does not introduce the subjectivity of hand-picking paradigm words. Using corpus set D[7], log entropy pre-processing and cosine distance as our distance metric, the most outlying word in a text is identified in the same way the previous classifier performed this step (because this step was performed well by the previous classifier [8]). The most outlying word's PMI score is then computed with each of the other content words in the text, rather than with paradigm words. Negative PMI scores would suggest that this outlying word does not normally co-occur with other words in the text and the magnitude of the scores would estimate the extent to which their tone differs. If the magnitude is large, and exceeds a certain threshold (the value of which is to be determined by tests such as this on the development set of texts), the outlying word would be considered significantly incongruous and the text would be regarded as a lexical register joke.

A test of the PMI method on the development set of texts was performed and when the lowest PMI score is used to determine whether a text is a lexical register joke or not, the highest accuracy achieved is 72.5% (see Table 4.5).

[7] Tests of the PMI method were performed before corpus set E was adopted.

[8] In 19/20 of the lexical register jokes, the most outlying word in the vector space was the word we, informally, selected as the most incongruous word in the text.

Table 4.5: Using lowest PMI score to decide if a text is lexical register joke or not (October 2009)

benchmark value	accuracy	precision	recall	F score
<=-8	28/40 (70%)	13/18 (72.2%)	13/20 (65%)	68.4
<=-8.5	29/40 (72.5%)	12/15 (80%)	12/20 (60%)	68.6
<=-9	26/40 (65%)	7/8 (77.8%)	7/20 (35%)	48.3

Table 4.6: Using average PMI score to decide if a text is lexical register joke or not (October 2009)

benchmark value	accuracy	precision	recall	F score
<= -5	33/40 (82.5%)	17/21 (81%)	17/20 (85%)	83
<= -6	32/40 (80%)	14/16 (87.5%)	14/20 (70%)	77.8
<= -6.5	33/40 (82.5%)	13/13 (100%)	13/20 (65%)	78.8

However when the average (rather than just the lowest) of the PMI scores is used to identify lexical register jokes within a collection of texts, results improve significantly. From Table 4.6 we see that an accuracy of 82.5%, a precision of 100% and a recall of 65% is achieved when a threshold of -6.5 is used. These results are comparable to those yielded by the vector space classifier described in Section 3.5 which uses corpus set D to build a vector space. That classifier yielded 80% accuracy, 73.1% precision and 95% recall. (See Table 3.9 for details).

Because the internet continually changes, a test of the latter PMI method (which uses the average of PMI scores) was performed on the same data four months later to determine how results might change with time. Results in Table 4.7 show that when the same threshold of -6.5 is used, accuracy, precision and recall scores universally drop. In fact a different threshold of -5 yields the best result of the later test (77.5% accuracy), but this is still lower than the best accuracy score achieved in the test performed just a few months earlier. The fact that the best threshold to use changes with time because of the constant flux of the internet, constitutes a major flaw in applying the PMI method to web data. It is possible that every time the PMI method were to be used on new data, a "test of the waters" could be performed using the development set to determine what the threshold should be, given the latest incarnation of the web. However the overhead of having to do this, and the inherent instability of applying this method to a growing and changing corpus, raises serious questions about the implementation described here. The 82.5% accuracy achieved in the October test and the 77.5% accuracy of the later test suggest that using co-occurrence data to detect lexical incongruities is promising, but it would be preferable if the corpus used to collect such data were not only large, but stable.

4.11 Conclusion

This chapter explored the extent to which corpus sparsity might account for classifier errors. Three ways of improving things were therefore investigated:

1. obtaining more corpora, (thereby creating a final corpora set E) so that fewer words with profile A or C appear.

Table 4.7: Using average PMI score to decide if a text is lexical register joke or not (February 2010)

benchmark value	accuracy	precision	recall	F score
<= -5	31/40 (77.5%)	16/21 (76.2%)	16/20 (80%)	78.1
<= -5.5	30/40 (75%)	14/18 (77.8%)	14/20 (70%)	73.7
<= -6	30/40 (75%)	13/16 (81.3%)	13/20 (65%)	72.2
<= -6.5	30/40 (75%)	11/12 (91.7%)	11/20 (55%)	68.8

2. performing Laplace smoothing

3. implementing a variation of the PMI method

Neither of the first two techniques significantly improved accuracy scores when tested on the development set of texts. Nevertheless the vector space delineated by corpus set E was adopted because it was thought that the extra data and coverage it provides might make the system better for generation and future testing with the test set.

The third method tried for improving the detection of lexical register jokes is a variation of the PMI method, and it yielded slightly better results. One test achieved 82.5% accuracy, the best classification result achieved so far. However results dropped to 77.5% accuracy when the very same test was performed 4 months later.

This is because content on the web (which is indexed by Altavista) is used by the system to predict whether a text is a lexical register joke or not, and this content is always changing. Using the web as a corpus for computing the probabilities of words co-occuring is tempting because of its enormous size, but the rapid growth and capriciousness of this resource might make our implementation of the PMI method too unstable to use.

This chapter also identified two types of words that are especially problematic for the vector space classifier to handle: words with profile A and profile C, as described in section 4.3. Even though words with profile A do not always appear as outliers in a text, they are nonetheless very difficult to interpret for the reasons given in section 4.3.1. Similarly words with profile C are problematic. Lexical register jokes sometimes make use of truly rare words to create opposition of tone but it would be naive to regard every text housing a rare word as a lexical register joke. Also, as stated in section 4.3.3, it is ambiguous in the first place whether words with profile C are in fact truly rare or only appear to be so because of corpus sparsity.

Increasing the size of the corpora and smoothing did not eliminate words with profile A or C - although the former step reduced their number. We have therefore decided that when it comes time for generation, texts containing words with these highly ambiguous profiles will be eliminated as candidate jokes.

Chapter 5

A better model of lexical register jokes

5.1 The structure of lexical register jokes

Our initial hypothesis was that in simple lexical register jokes, the tone of a single word opposes the tone of the rest of the text. At the end of Chapter 3, however, we made our description more precise by stating that the simplest kind of lexical register joke is a text in which the tone of a single word (tone A) conflicts with the tone of one or more other words in the text (tone B) and both tone A and tone B are presented lexically. Although not stated explicitly, this definition assumes that there are therefore three groups of words in lexical register jokes which are important to consider: a word with tone A (singleton set #1), words which are not involved in the conflict of tone and so can be considered neutral (set #2) and one or more words with tone B (set #3) which is somehow opposite to tone A.

Often there are multiple stopwords and very common words which have neutral tone within any given text and so set #2 will almost never be an empty or singleton set, although that is perhaps a possibility. Figures 5.1 and 5.2 show the different cases of simple lexical register jokes and Figure 5.3 represents more complicated lexical register jokes. The latter jokes are texts in which both set #1 and set #3 have more than one word in them.

In Chapter 3, one of the most successful classification algorithms searched for the most outlying word in a text and used its average distance with the rest of the words in the text to decide if a text is a lexical register joke or not. (Let us call this classifier #1). Given our 3 set model of the structure of lexical register jokes, this division of a text into only two groups - an outlying word and the rest of the words in a text - is overly simplistic and might obscure the extent to which an outlying word creates an opposition of tone. If the outlying word is set #1, then according to our more detailed model, classifier #1 has conflated sets #2 and #3 into a single group and measured the distance between set #1 and this overly large group when really only the distance between set #1 and set #3 should have been measured and used to estimate whether a text contains an incongruity of tone or not. In other words, neutral words (set #2) should probably not have been included in distance measurements as they could potentially dilute distance results and make important differences in tone less noticeable.

In spite of conflating sets #2 and #3 into a single entity and perhaps introducing error when it did so, classifier #1, as reported in the previous chapter, still managed to yield a respectable 77.5% accuracy in tests on the development set (using corpus set E). However we have implemented algorithms which align better with our more detailed model of lexical register jokes. The first

Figure 5.1: Simple lexical register jokes - case #1

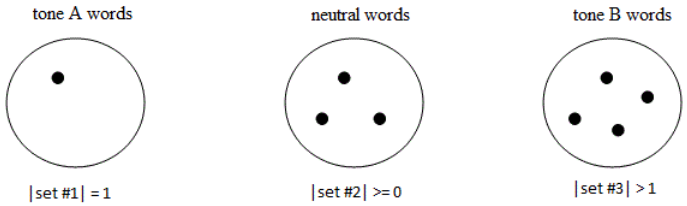
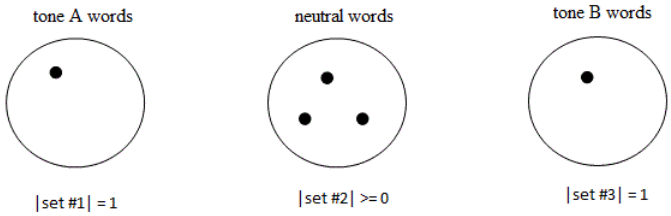


Figure 5.2: Simple lexical register jokes - case #2



algorithm clusters words into the three sets described above and measures the distances between these clusters. Our implementation and testing of this potentially more precise method is the main subject of this chapter.

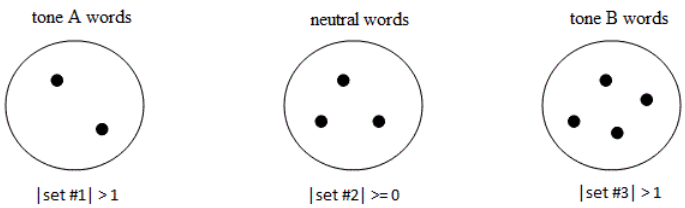
Another algorithm which emerged from the more detailed model - the fourth classifier proposed in this thesis - was also implemented and it too will be discussed in this chapter. This algorithm involves looking only at the distances between pairs of words and using the smallest of these to decide whether a text is a lexical register joke.

5.2 CLUTO's partitional clustering algorithm

Clustering algorithms can be divided into two classes: partitional and agglomerative solutions. Partitional clustering involves choosing random cluster centers and assigning each point in the dataset to the closest center. New centers are then computed - these are the averages of the points in a group - and a point is "moved to a new group if it is closer to that group's center than it is to the center of its present group" (Manly, 2004). This process is repeated until points no longer move to new groups. Agglomerative solutions, however, initially assign each point to its own singleton cluster. Pairs of clusters "are then successively merged according to some objective function until all points belong to the same cluster" (Kamvar et al., 2002).

In the past, "various researchers have recognized that partitional clustering algorithms are

Figure 5.3: More complicated lexical register jokes



well-suited for clustering large document datasets" (Zhao and Karypis, 2004). In the hope that this kind of clustering might also prove useful in our domain, we selected for our tests a partitional algorithm which is implemented in a software package called CLUTO (see Zhao and Karypis (2003) for details). This software was chosen because it implements a partitional algorithm but also because it outputs useful images of clustering results and provides multiple ways of defining the similarity and dissimilarity of objects in a vector space. Similarity and dissimilarity of objects are defined in the software by 'criterion' functions and the three functions used in our experiments are described below.

5.2.1 CLUTO's criterion functions

In CLUTO, criterion functions are divided into two classes: 'internal' functions which focus on intra-cluster similarity and 'external' functions, which focus on inter-cluster dissimilarity (Zhao and Karypis, 2004). In other words, internal criteria functions find clusters by focusing on minimizing the distance between words within a cluster whereas external criteria functions focus on finding clusters which are as far away from each other as possible. In our tests we experimented with two internal functions called I1 and I2 and an external function called E1.

In the formulas below, let the symbols u, v and w denote word vectors which have been normalized (i.e. have unit length) and the symbol S denote the set of n words we want to cluster[1]. Let S_1, S_2, ... and S_k denote the k clusters, n_1, n_2,... , n_k represent their sizes and *sim* represent the similarity function to be used for clustering. If S_i and S_j are two clusters of words containing n_i and n_j words respectively, let W_i, W_j and C_i, C_j be their corresponding composite and centroid vectors.

CLUTO's I_1 function is an internal criterion function which is defined below:

$$I_1 = \sum_{i=1}^{k} \frac{1}{n_i} \left(\sum_{v,u \varepsilon S_i} sim(v,u) \right)$$

In the experiments described in the previous chapter, the cosine function proved to be the best metric for outlier detection and classification so we decided to use it as the similarity function in all the clustering experiments. Therefore in the equation above, $sim(v,u)$ is replaced by $\cos(v,u)$ and this replacement is made directly in the other criterion functions described below.

CLUTO's I_2 function is another internal criterion function which is "used by the popular vector-space variant of the K-means algorithm" (Zhao and Karypis, 2004). Zhao and Karypis explain that "in this algorithm each cluster is represented by its centroid vector and the goal is to find the clustering solution that maximizes the similarity between each ... [point] ... and the centroid of the cluster that it is assigned to" (Zhao and Karypis, 2003). When the cosine metric is used to measure similarity between a point and a centroid, the criterion function to be maximized is:

$$I_2 = \sum_{r=1}^{k} \sum_{w_i \varepsilon S_r} \cos(w_i, C_r)$$

[1] In our experiments we do not normalize vectors to unit length before measuring their cosine distance. (Computing the cosine between two vectors actually performs this normalization). We assume that vectors are of unit length here so that the criterion functions can be simplified and perhaps understood better by the reader.

This formula can be rewritten as:

$$I_2 = \sum_{r=1}^{k} \sum_{w_i \varepsilon S_r} \frac{w_i^t C_r}{\| C_r \|} = \sum_{r=1}^{k} \frac{W_r^t C_r}{\| C_r \|} = \sum_{r=1}^{k} \frac{W_r^t W_r}{\| W_r \|} = \sum_{r=1}^{k} \| W_r \|$$

The third criterion function used in our experiments is an external criterion function called E_1 and it is defined as:

$$E_1 = \sum_{r=1}^{k} n_r \cos(C_r C)$$

where C is the centroid vector of the entire collection. The clustering algorithm which makes use of this function tries to "minimize the cosine between the centroid vector of each cluster to the centroid vector of the entire collection" (Zhao and Karypis, 2003). The equation for E_1which is being minimized, can be re-written as:

$$E_1 = \sum_{r=1}^{k} n_r \frac{C_r^t C}{\| C_r \| \| C \|} = \sum_{r=1}^{k} n_r \frac{W_r^t W}{\| W_r \| \| W \|} = \frac{1}{\| W \|} \left(\sum_{r=1}^{k} n_r \frac{W_r^t W}{\| W_r \|} \right)$$

where W is the composite vector of all the words in a text. (Since the initial term $\frac{1}{\|W\|}$is a constant, it can be excluded).

5.3 Classification tests using clustering

Tests were performed on the same set of development texts described previously, in order to determine how clustering compares to other classification methods tried so far. As in previous experiments, a word's normalized frequency counts in the set of corpora (corpora set E) are computed and log entropy pre-processing is performed on the data. In this way a multi-dimensional vector is constructed for each word in a text.

In previous experiments, the average cosine (λ) each word forms with the other words in the text was computed to find the most outlying word in a text. If the value of λ for the most outlying word was less than or equal to some threshold, the text was classified as a (simple) lexical register joke. (Recall that the smaller the cosine distance between two vectors, the further apart they are).

In our experiments with clustering however, classification of texts proceeds in a different manner. Word vectors are first of all grouped into clusters using CLUTO's partitional algorithm and the similarity between each of the clusters is computed. The similarity between clusters S_1 and S_2, for example, is computed by computing the average of the similarities between word w_i and word v_j for all w_i in S_1 and v_j in S_2.

Let's say a text consists of 5 words: personage, behest, would, to and computer. When we look for 3 clusters, a cluster with 'personage' and 'behest' is suggested (cluster #1 let's say), another consists of 'would' and 'to' (cluster #2) and a singleton cluster contains the word 'computer' (cluster#3). We compute the average cosine between a cluster and each of the other clusters. In other words, we compute the average of the cosines between

- personage & would
- personage & to

- behest & would
- behest & to

We then take the average of the cosines between

- personage & computer
- behest & computer

And finally we take the average of the cosines between

- would & computer
- to & computer

We use the smallest of these average similarities to judge whether the text is a lexical register joke or not: if it is less than or equal to some threshold, a cluster is considered significantly far away from another cluster, and the text is classified as a lexical register joke.

5.3.1 Factors varied

In our initial test, three factors were varied:

1. the joke boundary (defined in Section 3.5.2).
2. the criterion function used to cluster the data. (The functions I1, I2 and E1 were tried).
3. whether or not stopwords in a text are included.

The number of clusters to look for needs to be specified before the clustering program executes. In initial tests, we directed the software to search for three clusters because we believe this is the number of the different kinds of tone (a neutral tone and two conflicting tones) which occur in simple lexical register jokes. In later tests we will see how different numbers of clusters looked for by the classifier affect accuracy scores.

5.4 Results of CLUTO's partitional clustering

Tables 5.1 - 5.2 show the classification accuracy scores that result when we look for three clusters and the three factors listed above are varied. In the tests, eight joke boundaries spaced at intervals of 0.05 and ranging from 0.65 to 0.3 were used. However the tables exclude boundaries 0.65 and 0.6 as they produced consistently poorer results.

When stopwords are kept, the E1 function yields the highest accuracy score of 77.5% in distinguishing the development set of lexical register jokes from the original set of newspaper texts (i.e. not the newspaper quotes). When stopwords are excluded, the I1 and I2 criterion functions yield at best 72.5% accuracy and the E1 function yields 70% accuracy. All the scores resulting from different values of the three factors seem quite similar and it was difficult to determine which, if any of the factors had a significant effect on accuracy scores. We therefore performed a 3-way ANOVA on all the results falling within a joke boundary range of 0.5 to 0.3.

Table 5.1: k=3, keeping stopwords

joke boundary	E1 accuracy	I1 accuracy	I2 accuracy
<= 0.5	29/40 (72.5%)	30/40 (75%)	30/40 (75%)
<= 0.45	28/40 (70%)	28/40 (70%)	28/40 (70%)
<= 0.4	31/40 (77.5%)	28/40 (70%)	27/40 (67.5%)
<= 0.35	27/40 (67.5%)	30/40 (75%)	30/40 (75%)
<= 0.3	28/40 (70%)	28/40 (70%)	29/40 (72.5%)

Table 5.2: k=3, excluding stopwords.

joke boundary	E1 accuracy	I1 accuracy	I2 accuracy
<= 0.5	28/40 (70%)	29/40 (72.5%)	29/40 (72.5%)
<= 0.45	27/40 (67.5%)	28/40 (70%)	28/40 (70%)
<= 0.4	24/40 (60%)	24/40 (60%)	24/40 (60%)
<= 0.35	27/40 (67.5%)	26/40 (65%)	26/40 (65%)
<= 0.3	29/40 (72.5%)	29/40 (72.5%)	29/40 (72.5%)

5.4.1 Performing ANOVA

Performing ANOVA (Analysis of Variance) involves making a null hypothesis - which in our case is that all the scores come from the same distribution (i.e. they are not significantly different). A simple ANOVA test computes a probability (p) of whether the hypothesis is true or not and a commonly used significance level is 0.05. If, for example, we compare test scores from school A with test scores from school B and ANOVA returns a p value of 0.05 or less, this suggests that the two sets of scores are significantly different. A p value of 0.05 means that there is only a 5% chance that test scores this different would arise from randomly selecting from the same (normal) distribution.

The more complicated 3-way ANOVA which we performed on the test results returns three values because it computes a probability (p) for the three factors we are interested in:

1. factor #1: the joke boundaries

2. factor #2: the criterion functions

3. factor #3: whether or not stopwords in a text are kept.

N-way ANOVA estimates how much variance in the data is due to a given factor and computes the probability of that variance resulting simply from chance. In other words, for each factor, a probability is computed and that probability is whether differences in scores arising from different values of a factor are significant or not.

The 3-way ANOVA on the classification accuracy scores returned p values of 0.0203 for factor #1, 0.9031 for factor #2 and 0.0048 for factor #3. In other words the test concluded that factors #1 and #3 have a significant effect on accuracy scores.

Figure 5.4: Multiple comparison test for different values of factor #1

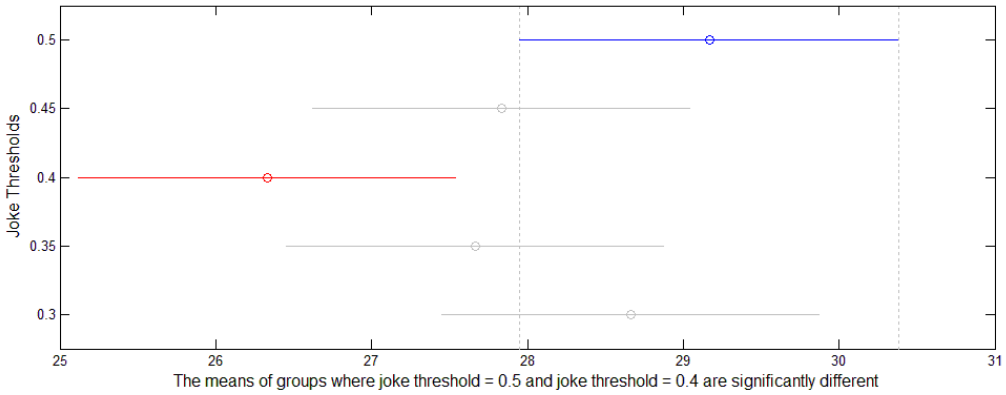
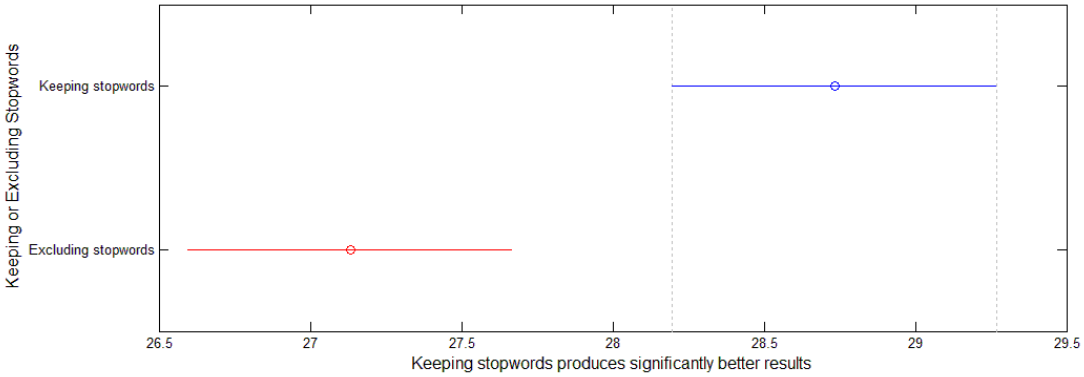


Figure 5.5: Multiple comparison test for different values of factor #3



5.4.2 Multiple comparison testing

ANOVA, however, only suggests whether groups of data are significantly different - it does not indicate which groups are actually different and produce better results. To determine this, a multiple comparison test was performed.

Figures 5.4 and 5.5 show which values for factors #1 and #3 yield significantly different accuracy scores. Figure 5.4 shows that the value for factor #1 which yields the highest mean accuracy is 0.5. The exact value of this mean is 29.17 and it is represented as a circle in the middle of the top line in Figure 5.4. In multiple comparison testing, means are significantly different only if their confidence intervals, which appear as horizontal lines in the figures, are disjoint[2]. Thus from the figure we see that the mean accuracy score for a joke boundary of 0.5 yields the highest mean accuracy score but is only significantly higher than the mean accuracy scores arising from a boundary of 0.4.

[2]With a certain level of confidence, we predict that the population mean appears somewhere within the range demarcated by the confidence interval. The confidence intervals pictured in the figures are 95% confidence intervals. This means that if you were to take 100 samples from a population distribution and to build confidence intervals for these samples, 95 of them would contain the population mean. See Hinton (2004) for details.

Similarly, Figure 5.5 shows that, according to the multiple comparison test, keeping stopwords produces a mean accuracy score (28.73) which is significantly higher than the mean accuracy score that occurs when stopwords are excluded (27.13). In our final model, we will therefore resolve to keep stopwords.

Given this decision, we decided to perform a 2-way ANOVA on the results that arise when stopwords are kept to see whether factors #1 and #2 had significant effects on scores when factor #3 is fixed in this way. This 2-way ANOVA returned p values of 0.6598 for factor #1 and 0.9671 for factor #2. The test suggests that which joke boundary and which criterion function to use when stopwords are kept does not matter - none of the different values for these factors yielded significantly higher accuracy scores. We will wait, however, until we have performed further testing before we decide which joke boundary and which criterion function to use in our final classification algorithm. (Recall that we are fine-tuning our classification method here using development data. Once this formative work has been done, our model will be tested on unseen lexical register joke data).

5.5 Results of clustering (using newspaper quotes)

When newspaper quotes are used as the development set of regular texts, classification tests yield the accuracy scores listed in Table 5.3.

Table 5.3: Classification results on newspaper quotes, k=3, keeping stopwords

joke boundary	E1 accuracy	I1 accuracy	I2 accuracy
<= 0.65	33/40 (82.5%)	33/40 (82.5%)	33/40 (82.5%)
<= 0.6	37/40 (92.5%)	37/40 (92.5%)	37/40 (92.5%)
<= 0.55	36/40 (90%)	35/40 (87.5%)	35/40 (87.5%)
<= 0.5	33/40 (82.5%)	34/40 (85%)	34/40 (85%)

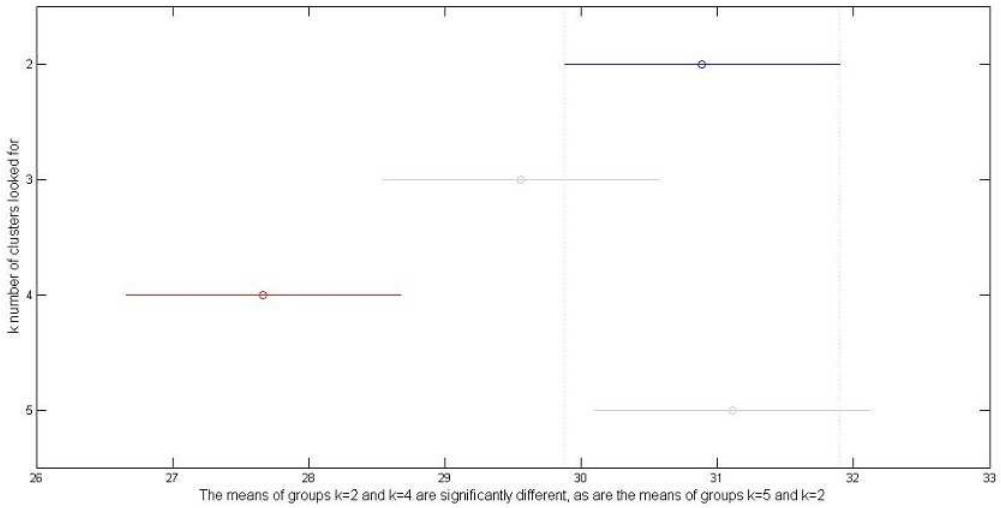
The I1 and I2 accuracy scores for the boundaries shown are identical and the E1 scores are nearly so. The best accuracy score for all three criterion functions is 92.5%: a high score but possibly misleading because, as stated previously, the development set of newspaper quotes may be uncharacteristically bland.

5.6 Looking for a different number of clusters

We wanted to see what sort of accuracy scores result when the clustering algorithm looks for a different numbers of clusters other than three. We felt this might be important because it may be that our intuition is wrong about the words of a lexical register joke naturally dividing into three groups according to their tone. Another possibility is that our hypothesis about the number of clusters to look for is sound but the vector space we have built is flawed in such a way that a non-intuitive value of k might prove to be more useful when classifying texts. If the vector space is a flawed estimate of tone, but is not useless in this regard (after all a classification accuracy of 77.5% was achieved on the development set when k equals three), then a more empirical approach to choosing a value of k might be called for.

We therefore determined what kinds of accuracy scores result when k equals 2, 3, 4, and 5 and compared these four sets of scores using a 1-way ANOVA. For a given value of k, the top

Figure 5.6: Multiple comparison test of scores resulting from different values of k



three scores for each criterion function were collected. This resulted in a set of 9 scores (a triad of scores for E1, I1 and I2) and these 9 scores represented the (best) scores yielded for a given value of k.

The 1-way ANOVA returned a p value of 0.1189 which is above the significance level of 0.05. Thus accuracy scores resulting from different values of k are not significantly different. The k=3 scores are not significantly worse or better than any of the other results and so a more empirical choice of k is not to be preferred over our theoretical value of k=3. A reason it does not matter which value of k is used may have to do with the fact that we are testing simple lexical register jokes in which a single word is in opposition to one or more other words in a text. For different values of k, the most outlying word still appears in its own cluster (or has at most 1 other word in its cluster) and this word is roughly the same distance from the furthest cluster, regardless of how many other clusters are found by the algorithm.

5.7 Where did classification go right and wrong?

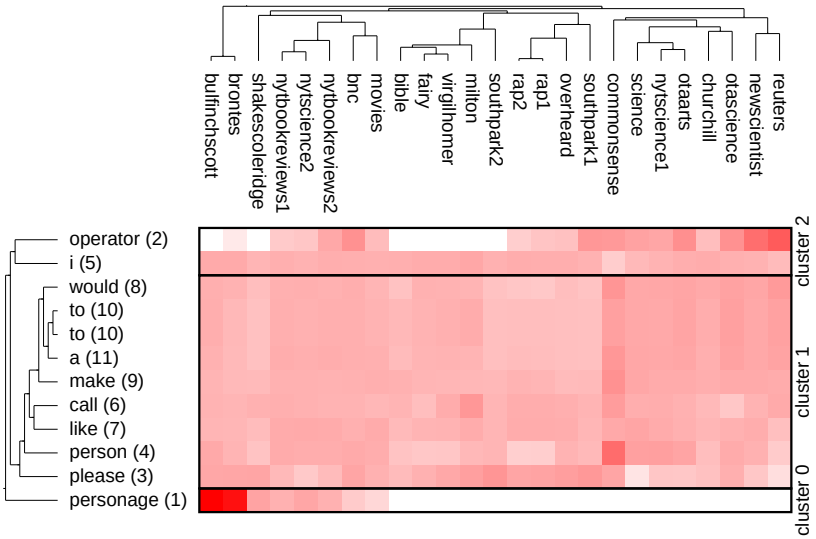
When k=3 and the criterion function is I1 for example, six newspaper texts and four lexical register jokes were improperly classified. We will speculate about why these mistakes were made and examine some examples of successful classification. But first we will describe the various pieces of information which the classification algorithm outputs upon analysing a text.

5.7.1 Explaining the output of the classification program

Figure 5.7 shows the frequency count patterns of the words in lexical register joke #1 and how the classifier clusters these words. The grey cells in the figure represent the frequencies of words (rows) in various corpora (columns): the darker the cell, the higher the frequency. (These frequencies have been transformed by log entropy processing). The horizontal lines dividing the figure represent the clusters. Thus the word ‘personage’ appears in its own cluster (which is labelled cluster 0 on the right hand side of the diagram), cluster 1 consists of mostly stopwords and cluster 2 contains the words ‘operator’ and ‘i’.

The hierarchical tree on the left side of Figure 5.7 shows which words are most closely

Figure 5.7: k=3 clustering for lexical register joke #1

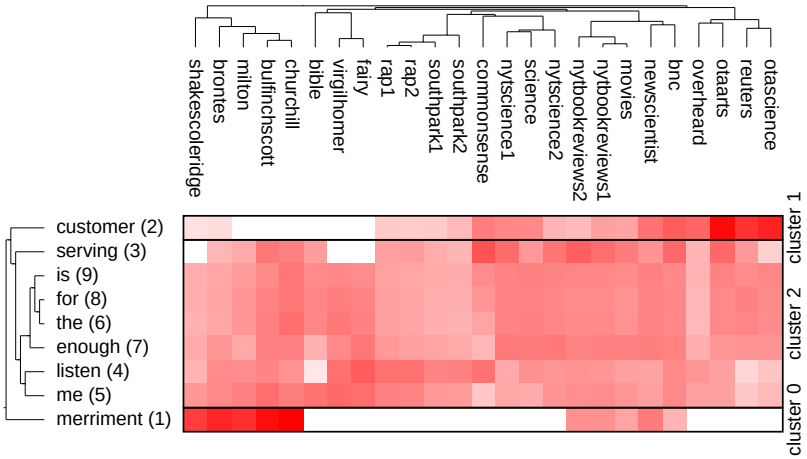


related to each other. For example the word 'to' appears twice in the text and so the two instances of this word are the first to cluster (in an agglomerative solution) because they are identical. The stopwords 'a' and 'would' are the next pair to merge and they then combine with the first pair. The last word to join the hierarchical tree is the incongruous word 'personage' because it is the most outlying word. We used a partitional algorithm to cluster texts but the hierarchical tree we are describing here was produced "by performing a hierarchical agglomerative clustering on top of" the partitional algorithm (Zhao and Karypis, 2003). An option can be specified in CLUTO (the part of the classifier which clusters the words of a text based on their frequency counts) which "builds a complete hierarchical tree that preserves the clustering solution that was computed", even if that solution was produced by a partitional algorithm (Zhao and Karypis, 2003). We selected this option because it shows at an atomic level which words and groups of words in a text are most similar and dissimilar.

Similarly a hierarchical tree appears at the top of Figure 5.7 and it demonstrates which corpora are most alike in terms of the words of a single text (i.e. which corpora display similar frequency count patterns). Using so few frequency count patterns (only twelve in the case of lexical register joke #1 for instance because it contains only twelve words) to draw any conclusions about the corpora would be unwise and so this information was largely ignored - it was certainly not used by the classification algorithm to decide whether a text is a lexical register joke or not.

Returning to Figure 5.7, the numbers which appear in brackets after the words are the rank of the words, as described in the previous chapter, and have nothing to do with CLUTO's clustering of the data. We include this ranked list information in Figure 5.7 and in the other figures below to see if the outlier word detected by the earlier method appears in its own cluster when clustering is performed. In the outlier detection method described previously, a word's cosine distance from each of the other words in a text was calculated and the distances added together. This sum was computed for each word and in this way a ranked list was produced. The word at the top of the list was furthest away from the others and was therefore considered the one most likely to be incongruous.

Figure 5.8: Classifier's clustering of lexical register joke #13



In the case of the joke shown in Figure 5.7 (lexical register joke #1) and 13 other simple jokes from the development set, the words previously identified as outliers appear in singleton clusters. In the 6 remaining jokes, the outlier word appears in a cluster with only one other word. Thus there appears to be a correspondence between the outputs of the outlier detection method described earlier (in Chapter 3) and the clustering algorithm described here. This is perhaps unsurprising given that the cosine metric is used in both cases as a way of estimating similarities between words - although in the case of clustering, the cosine metric is incorporated into a criterion function rather than used alone as a measure of similarity.

5.7.2 Comparison of intuitive and automated clustering

In order to avoid any bias, texts of the development set were clustered by hand before looking at how the classifier performed this task. (Appendix C shows how we clustered the words of all the jokes in the development set). This section compares how the classifier clustered texts to how we grouped the words of lexical register jokes according to their tone.

5.7.2.1 Jokes where classifier and our intuition agree

Five jokes (jokes 7, 11, 13, 17, and 18) were clustered by the classifier exactly the way we clustered them by hand. For example Figure 5.8 shows that the classifier put the words 'customer' and 'merriment' into different singleton clusters and the rest of the words in a third cluster and this is the way we clustered the text according to our intuition.

5.7.2.2 Jokes whose words were clustered nearly the same way as our intuition

Seven jokes in the development set (jokes 1, 2, 4, 6, 9, 14, 20) were clustered in nearly identical ways to how we grouped words of the text based on the formality of their tone. Take for example joke 1: 'Operator, I would like to make a personage to person call'. Figure 5.7 shows the frequency count patterns of the words in this text and how the words were clustered. We find that apart from the stopword 'I' appearing with the word 'operator' in cluster 2 rather than with the other stopwords of cluster 1, the clustering solution for lexical register joke 1 is identical to how we clustered the words of this text by hand. Table 5.4 shows how we clustered the words of lexical register joke 1 by hand and compares this to how the classifier clustered the text.

A reason the word 'I' does not appear in cluster 1 might be because it appears relatively

infrequently in the common sense corpus (it appears only 209 times) whereas the other stopwords appear 3 to 382 times more frequently in that corpus. Sentences in the common sense corpus tend not to be subjective statements (some examples are: "Eyes are used to see things", "A bird can fly", "Most people sleep in a bed.") and perhaps the relatively low frequency of the word 'I' in this single corpus is enough to set this word apart from the other stopwords. Judging from appearances, the tile pattern or 'fingerprint' for the word 'i' seems much more similar to the fingerprints of the other stopwords - certainly more similar to these patterns than the one for the word 'operator' - yet the cold facts of the cosine measurements disagree with this judgement.

Table 5.4: Comparing clustering by hand and by classifier (lexical register joke #1)

tone	group 1	group 2	group 3
how we clustered	personage	i, would, like, to, make, a, person, call	operator
how the program clustered	personage	would, like, to, make, a, person, call	operator, i

The algorithm's clustering of words in this text accords almost perfectly with how we would cluster the data and the distance between the singleton cluster 0 and the rest of the words in the text is large enough for the text to be considered a lexical register joke.

The algorithm's clustering of joke 9 also closely resembles how we clustered the words of this text (see Table 5.5). The only difference is that the clustering algorithm included the word 'oh' in the neutral cluster and the word 'tickets' in the informal cluster whereas we put 'oh' in the informal cluster and 'tickets' in the neutral cluster. More importantly, the formal word 'crestfallen' formed a singleton cluster which was significantly far away from the 'informal' cluster ('gee', 'fellas', 'gonna' etc.) in the vector space and the text was classified as a lexical register joke.

Table 5.5: How we clustered lexical register joke #9

tone	group 1	group 2	group 3
how we clustered	crestfallen	how, could, I, fall, for, the, are, be, tickets	fake, Superbowl, gee, fellas, gonna, oh
how the program clustered	crestfallen	how, could, I, fall, for, the, are, be, oh	fake, Superbowl, gee, fellas, gonna, tickets

The five other lexical register jokes which were clustered in nearly identical ways to how we grouped words of the text are similar to the two examples above. The word with tone A forms a singleton cluster and either

1. all the words we identified as having tone B appear in another cluster but a word we classified as neutral also appears with them

2. all the words we identified as having tone B appear in another cluster except for one, which appears in the neutral cluster.

5.7.2.3 Joke where clustering is quite different from intuition, but not obviously wrong

Joke 10 was particularly difficult to cluster by hand and so determining whether the classifier's clustering of the words in these texts was inferior to our own is ambiguous. In this text, the word 'gee' is informal and is the only word in the text to contribute to that end of the opposition (i.e. it has tone A). Exactly which of the other words in the text oppose its informality of tone (i.e. words with tone B) is less clear, however. As Table 5.6 suggests, we felt that the words 'potential', 'simultaneously', 'clients', 'annual', 'yield', and 'creative' were business-like words which oppose the informal tone of the word 'gee'.

When the classifier processed this text, it also placed the word 'gee' in a singleton cluster and grouped all the words we identified as having a formal business-like tone in another cluster, except for the word 'yield', which it put in the neutral cluster. However the classifier also added the words 'reasonable', 'provide', 'growth', 'issues', and 'determining' to the cluster of words we identified as business-like. These words are not obviously wrong selections however - many of them do in fact convey a formal business-like tone - and it is difficult to know whether the classifier's clustering of the text is actually better or worse than the clustering we performed by hand.

Table 5.6: How we clustered lexical register joke #10 compared to the classifier

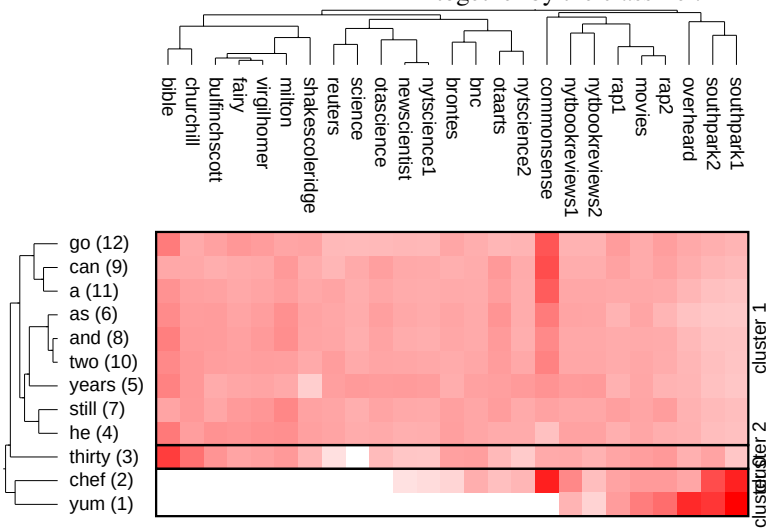
tone	group 1	group 2	group 3
how we clustered	potential, simultaneously, clients, annual, yield, creative	determining, which, issues, have, growth, while, working, to, provide, your, with, a, reasonable, is, most, certainly	gee
how the program clustered	potential, simultaneously, clients, annual, creative, reasonable, provide, growth, issues, determining	which, have, while, working, to, provide, your, with, a, is, most, certainly, yield	gee

5.7.2.4 Jokes where clustering is significantly different from intuition (and obviously wrong)

The classifier clustered 7 joke texts in significantly different ways than we did. In fact in 5 of these texts, pairs of words which we believed possess opposite kinds of tone, were placed in the same cluster by the classifier. These pairs of words were 'chef' and 'yum' (joke 3), 'moola' and 'gentlemen' (joke 5), 'damn' and 'mommy' (joke 12), 'floozies' and 'parameters' (joke 16) and 'triumvirate' and 'twinkies' (joke 19). Figure 5.9 shows the classifier's incorrect pairing of opposite words in joke #3.

Upon closer analysis, however, the words 'chef' and 'yum', which we initially felt created

Figure 5.9: Words with opposite tone in lexical register joke #3 were incorrectly clustered together by the classifier.

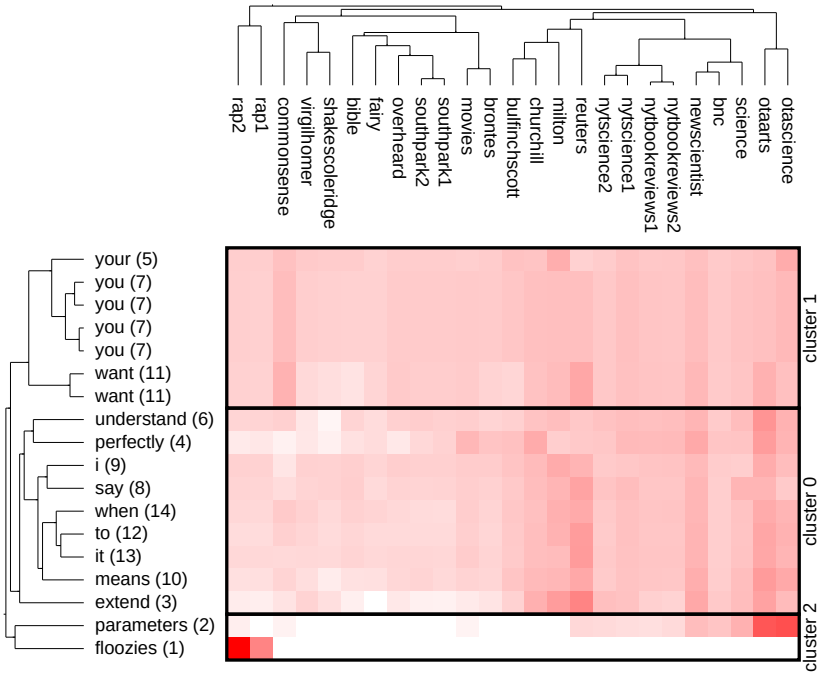


an opposition of tone in joke #3, do not seem to have significantly different kinds of tone and this text ('Thirty two years as a chef and he can still go yum'), which is a caption taken from a New Yorker cartoon, is perhaps the weakest lexical register joke in the development set. This is because details of the cartoon from which this caption is taken make an important contribution to the opposition of tone occurring in this joke, but this contribution is lost when the caption is divorced from its cartoon. The cartoon for this caption shows a chef talking to a waiter about another chef, presumably the head chef of a kitchen, who can be seen in the background of the cartoon sampling a dish. Both chefs are wearing formal uniforms – the traditional white chef's coat and hat - and the waiter is also dressed formally in a tuxedo. These details suggest that the setting of the joke is the kitchen of an expensive restaurant and so a somewhat formal context is created by the cartoon, which is then undermined when the informal word 'yum' appears in the caption.

When we initially reviewed this text to decide whether it was a lexical register joke or not, we believed that the tone of the word 'chef' (and no other word in the text) was able to connote the formality of the setting when the caption is read without its cartoon. This judgement seems wrong in retrospect and the tones of the words 'chef' and 'yum' do not seem significantly different or opposite. The classifier's placement of the words 'chef' and 'yum' in the same cluster may therefore not be a serious error – the problem here may have more to do with our inclusion of this text into the development set of jokes rather than with the classifier's clustering algorithm. (To avoid mistakes such as this occurring in the test set of lexical register jokes, a filtering step using volunteeers was performed. See Chapter 6 for details).

Although clustering together the words 'chef' and 'yum' from joke #3 may not be a gross error on the part of the classifier, examples can be found in which it mistakenly placed words with more clearly opposite tone into the same cluster. For example the words 'parameter' and 'floozies', which we identified as opposites by hand, appear in the same cluster when the classifier processed joke #16. From their frequency count distributions, shown in Figure 5.10, we find that what these two words have most in common is perhaps not that they appear in many of the same

Figure 5.10: Frequency count distribution and clustering of joke #16



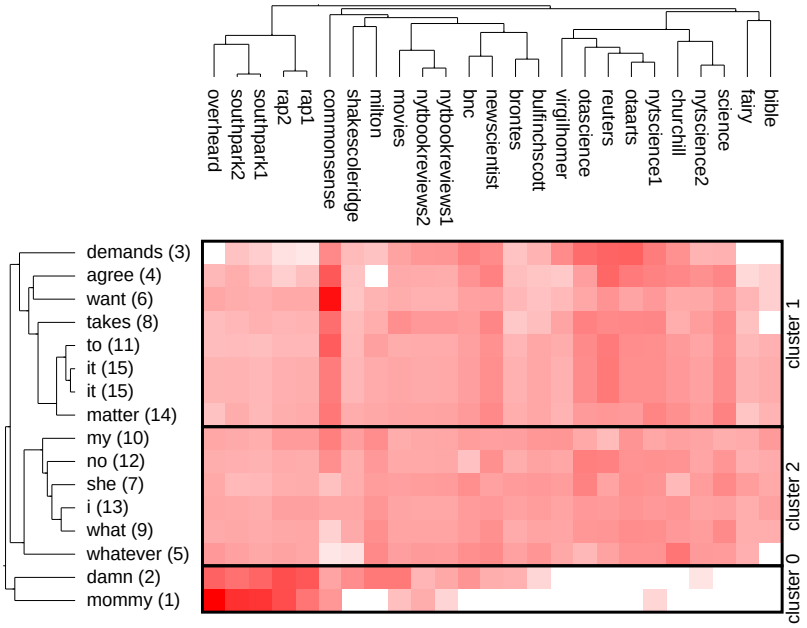
corpora – they only have 1 corpus in common – but that, of all the words in the text, they appear in the fewest number of corpora. Because all of the other words in the text, without exception, appear in all of the corpora, these two words, which do not, are regarded as similar. This again highlights the fact that when numerous words in the text appear in most of the corpora, words that do not appear in many corpora, but have little else in common, are often identified as alike by the classifier.

Joke 19 is similar to joke 16. Two words which we considered as having opposite tone - 'twinkies' and 'triumvirate' - only have a single corpus in common, but both are the only profile A words to appear in the text, and perhaps for this reason, they are placed in the same cluster by the classifier.

Similarly, two words in joke 12, which we identified by hand as having opposite tone, were clustered together by the classifier. This joke differs from the other jokes in the development set, however, because the two opposite words housed within it ('damn' and 'mommy') are not opposite in the usual way - where one word is formal and the other is informal. Instead, both of these words possess different kinds of informal tone: 'damn' is a curse and therefore harsh (and swear words tend to be regarded as informal uses of language) while the word 'mommy' is child-like. The vector space, however, is not subtle enough to distinguish between differing kinds of informality. Because both words are informal, they appear in many of the same corpora (as we can see in Figure 5.11), have similar frequency vectors in the space, and so are considered alike rather than different.

Jokes 8 and 15 were also clustered in significantly different ways than the way we clustered them by hand. In these cases, the classifier did not place words which were identified as opposites by hand, into the same cluster, but the clustering suggested by the classifier is nonetheless significantly different from our intuitive grouping of words. Why the classifier clustered the words of

Figure 5.11: Words (from joke 12) with different kinds of informal tone are clustered together



these texts so differently than we did is unclear.

5.7.3 Incorrectly classified newspaper texts

The six newspaper texts which were incorrectly classified as lexical register jokes were:

- *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...*
- *at thursday night performance the boy beside me could not have been more than actually **shrieked** in*
- *both runs were completely sold out and he was so **mobbed** at the stage door that he*
- *advertising executives packed into the hall for television presentation a **rite** of spring passage in the world*
- *he **contorts** a bit raising his right shoulder wringing one hand with the other and fingering his ...*

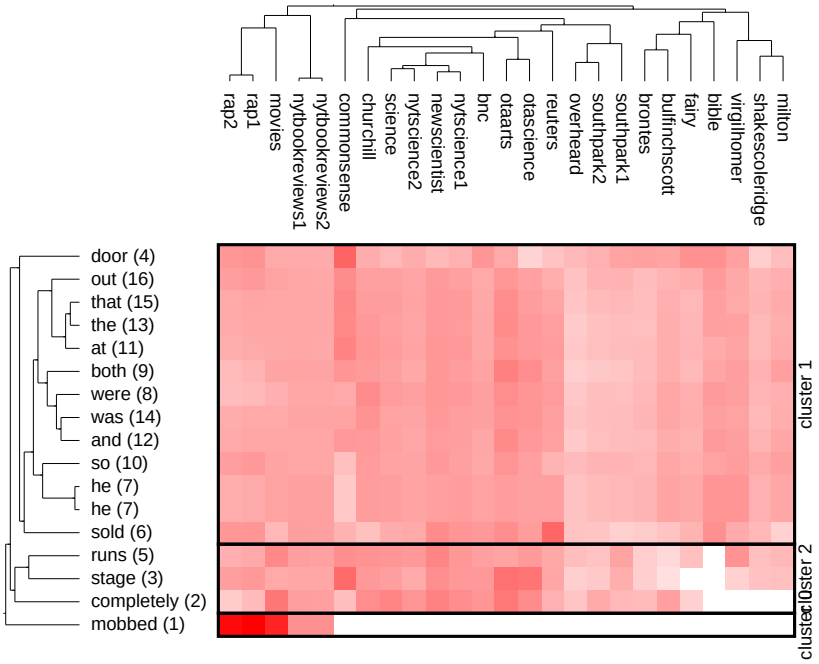
These same texts (along with 2 others) were mistakenly classified as lexical register jokes by the classifier described in chapter 4 (classifier #1). This is perhaps not surprising when we recall from Section 5.1 how similar are classifier #1 and the clustering classifier described in this chapter (let's call this classifier #3) [3]. Further evidence of their similarities is found when we notice that each of the words shown in bold above were considered the most outlying words by classifier #1 and when

[3]Classifier #2 is the classifier described in Chapter 3 which uses a variation of the PMI method to estimate the similarity of the tones of words.

the clustering classifier executed its algorithm, these same bolded words were placed in singleton clusters.

The six texts listed above highlight what is perhaps the biggest problem with classifier #3 (and continues to be a problem plaguing classifier #1): texts containing profile A words are almost always classified as lexical register jokes. If we define profile A (and D) words to be words that occur in six or fewer corpora then all of the words shown in bold, except for one (the word 'rite') are profile A words. None of the other words in the newspaper texts appear in so few corpora and when the number of clusters looked for by the classifier is equal to 3, these profile A words all form singleton clusters. Figure 5.12 for example demonstrates how pronounced the difference is between the word 'mobbed' and the rest of the words in newspaper text 17. We believe that this difference is inflated and has more to do with the sparsity of the corpus data than with the word 'mobbed' being truly esoteric. Because the other words in the text - content words and stopwords - appear in a relatively large number of corpora (none of the other words appear in fewer than six corpora), the distance between their vectors and the vector for the word 'mobbed' is unduly large and the text is thus mistakenly classified as a lexical register joke. Although the amount of corpus data was enlarged (to corpus set E), corpus coverage still seems to be inadequate and continues to hamper classifiers #1 and #3.

Figure 5.12: Newspaper text #17 from the development set



5.8 A classifier (#4) which looks at cosine pairs

Figures 5.1 to 5.3, from section 5.1 show the different kinds of lexical register joke. In each case, at least one pair of words creates an opposition of tone. There may be other words involved in creating oppositions of tone but it is enough to identify just one of these oppositions. An algorithm which computes the distances between each pair of words in a text and uses the lowest of these to decide if a text is a lexical register joke should therefore suffice as a classifier. In fact such an algorithm might perform better than classifier #1 (the average cosine method). Figure 5.13 shows a

Figure 5.13: Looking for largest pair-wise distance may be better than computing average distances

1 dimensional space in which 5 words from a text (items A to F in the diagram) have been plotted. In the figure, words A and F are significantly far apart, suggesting that an incongruity of tone exists between this pair of words. This fact may be lost, however, when classifier #1 computes, for instance, word A's average distance to all the words in the text. As the figure suggests, word A is quite close to words B and C in the space. When these small distances, along with word A's distances to the other words in the text are added together and an average taken, the significantly large distance between words A and F will have been watered down in the resulting mean[4]. If both words A and F have such close neighbours, their average distances to other words in the text might not exceed the joke boundary and the text would be misclassified as regular text.

A classifier (classifier #4) that computes the cosine distances between every pair of words in a text and uses the lowest of these to decide if a text is a lexical register joke was therefore implemented and Table 5.7 shows its performance when classifying the development set of texts.

Table 5.7: Lowest cos method on development set

joke boundary	accuracy	precision	recall	F score
<= 0.2	26/40 (65%)	9/12 (75%)	9/20 (45%)	56.3
<= 0.25	31/40 (77.5%)	14/17 (82.4%)	14/20 (70%)	75.7
<= 0.3	29/40 (72.5%)	14/19 (73.7%)	14/20 (70%)	71.8
<= 0.35	30/40 (75%)	16/22 (72.7%)	16/20 (80%)	76.2
<= 0.4	30/40 (75%)	17/24 (70.8%)	17/20 (85%)	77.3
<=0.45	28/40 (70%)	17/26 (65.4%)	17/20 (85%)	73.9

A joke boundary of 0.25 yielded the highest accuracy of 77.5%. Although this classifier should theoretically perform better than classifier #1, in practise it did not improve on the accuracy of classifier #1.

5.9 Hybrid classifier

A fifth and final classifier which is a hybrid of classifiers #3 and #4 was also implemented. This classifier requires that both classifiers #3 and #4 agree that a text is a lexical register joke if the text is to be classified as such. Adding this requirement could reduce the number of false positives proposed by the classifier and might therefore improve on the accuracy and precision scores of

[4]The vector space represented in Figure 5.13 is a simplified 1 dimensional space - the actual vector space used by the algorithm has 15 dimensions - but the argument proposed in this section holds for spaces with higher dimensions.

classifiers #3 and #4 when these are run alone. (On the other hand, the hybrid could also have a detrimental effect on all the scores. For instance recall scores of the hybrid will always be equal to or lower than the lowest recall score of classifiers #3 or #4, given that more restrictions have been imposed on a text for it to be considered a lexical register joke).

Table 5.8: Hybrid scores on development set (using I1 criterion function, keeping stopwords)

classifier	accuracy	precision	recall	F score
classifier #3 (clustering, I1 function, k=3, boundary <= 0.5)	30/40 (75%)	16/22 (72.7%)	16/20 (80%)	76.2
classifier #4 (lowest pairwise cos with boundary of 0.25)	31/40 (77.5%)	14/17 (82.4%)	14/20 (70%)	75.7
classifier #5 (hybrid of classifiers #3 and #4 above)	31/40 (77.5%)	14/17 (82.4%)	14/20 (70%)	75.7

Table 5.8 demonstrates that the hybrid classifier does not improve on the highest accuracy or precision scores of classifiers #3 and #4 (which are shown in rows 1 and 2 of the table) when tested on the development set. In fact it achieves exactly the same results as classifier #4.

5.10 Summary of the 5 classifiers

As Table 5.9 suggests, all five classifiers tested up to this point achieved remarkably similar results. Because no one stands out, all five of the classifiers will be evaluated on the test set.

5.11 Classifying more complicated lexical register jokes

Classification which uses clustering (classifier #3) and classification which looks at pairwise cosines (classifier #4) did not improve on the accuracy scores of classifiers #1 (average cosine method) or #2 (the classifier that uses a variation of the PMI method to estimate tone similarity). An advantage of classifiers #3 and #4 over these others, however, is that they should be able to recognize more complicated lexical register jokes. Recall from section 5.1 that in more complicated lexical register jokes, the tone of groups or clusters of words stand in opposition to each other, as opposed to simple lexical register jokes in which only a single word opposes the tone of one or more other words in a text. Clustering algorithms could potentially discover clusters of words which are far apart from each other in a vector space, thus providing a way of detecting more complicated kinds of lexical register jokes. Similarly, the lowest cosine method should also be able to detect more complicated lexical register jokes. All lexical register jokes, regardless of their complexity, contain at least one pair of words whose tones are in opposition. Classifier #4 simply looks in a text for the pair of words that are furthest apart in the vector space and classifies the text based on that information. In this way it too should be able to detect the whole class of

Table 5.9: Summary of the 5 classifiers' performance on the development set

classifier	accuracy	precision	recall	F score
classifier #1 (avg cos method) <= 0.6	31/40 (77.5%)	19/27 (70.4%)	19/20 (95%)	80.9
classifier #2 (PMI method, threshold <= -5 October 2010 result)	31/40 (77.5%)	16/21 (76.2%)	16/20 (80%)	78.1
classifier #3 (clustering, I1 function, k=3, boundary <= 0.5)	30/40 (75%)	16/22 (72.7%)	16/20 (80%)	76.2
classifier #4 (lowest pairwise cos with boundary of 0.25)	31/40 (77.5%)	14/17 (82.4%)	14/20 (70%)	75.7
classifier #5 (hybrid of classifiers #3 and #4 above).	31/40 (77.5%)	14/17 (82.4%)	14/20 (70%)	75.7

lexical register jokes.

Assembling a set of more complicated lexical register jokes with which the classifiers might be tested, proved difficult however: only three were found in a manual search of over 11,000 New Yorker cartoon captions and a fourth example was found in the movie Monty Python and the Holy Grail. These four more complicated lexical register jokes are:

1. Would you care to know dear that your hazy **morn** of an **enchanted** day in May is composed of six tenths parts per million sulfur dioxide, two parts per million carbon monoxide, four parts per million hydrocarbons, three parts ...

2. And then he strode into the board meeting **brandishing** this flaming sword and said **Woe unto** him who moves his corporate headquarters out to the suburbs.

3. And finally **Lord** may **thy** wisdom guide and inspire the deliberations of those **thy** servants our planning and zoning commission.

4. Oh, **Lord**, bless this **thy** hand grenade that with it **thou mayest** blow **thy** enemies to tiny bits, in **thy** mercy.

In each of the texts, words in bold communicate one kind of tone and words which we believe convey a conflicting tone are underlined. Given the difficulty of finding examples of this kind of lexical register joke, we were unable to create both a development and test set of examples. In fact four examples hardly constitutes even a single viable set with which we might test the performance

of an algorithm. Nevertheless, it seemed worthwhile to determine if classifiers #3 and #4 are better able to recognize this type of joke than the first two classifiers, which were developed to recognize only the simplest kind of lexical register joke. A test set consisting of these 4 more complicated lexical register jokes and the same set of 20 regular newspaper texts from the development set was therefore created.

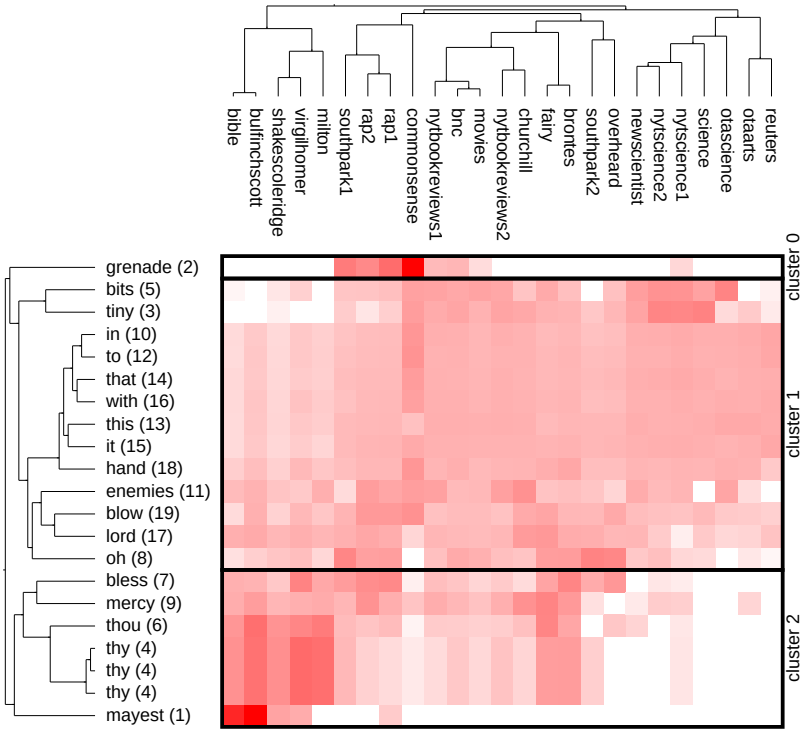
Table 5.10: Performance of the classifiers on more complicated lexical register jokes

	accuracy	precision	recall	F score
classifier #1 (avg cos method) <= 0.5	18/24 (75%)	3/8 (37.5%)	3/4 (75%)	50
classifier #2 (PMI method)	17/24 (70.8%)	3/9 (33.3%)	3/4 (75%)	46.1
classifier #3 (clustering I1 function, keeping stopwords) <=0.51	17/24 (70.8%)	3/9 (33.3%)	3/4 (75%)	46.1
classifier #4 (lowest pairwise cos) <=0.25	21/24 (87.5%)	4/7 (57.1%)	4/4 (100%)	72.7
classifier #5 (hybrid) <= 0.51 for esim, 0.25 for lowest pairwise cos	20/24 (83.3%)	3/6 (50%)	3/4 (75%)	60

Table 5.10 shows how all five classifiers performed on the test set. Classifier #1 achieved an accuracy of 18/24 (75%) when a joke boundary of 0.5 was used. 15/20 newspaper texts were correctly classified and 3/4 of the jokes were correctly identified as jokes. Classifier #2, which uses a variation of the PMI method and goes out on the web to compute co-occurrence statistics, yielded 17/24 (70.8%) accuracy. 14/20 newspaper texts and 3/4 of the jokes were correctly classified. Interestingly classifier #3, which theoretically should be better at recognizing more complicated lexical register jokes than classifiers #1 and #2, performed no better than these – in fact it achieved the same accuracy as classifier #2. The test did provide some indication, however, that classifier #3 may be grouping together words with similar tone. For example Figure 5.14 shows that the classifier's clustering of words in joke #1 is quite close to our intuitive clustering of the text into clusters that have archaic, modern and neutral tones.

Classifier #4 produced the best results of all. It correctly classified 17/20 newspaper texts and 4/4 of the jokes. Its accuracy of 21/24 (87.5%) is 12% to 16% higher than the other classifiers. This seems like a marked improvement, but when we consider the smallness of the development set, which consists of only 4 complicated lexical register jokes and 20 newspaper texts, an improvement of 12% to 16% accuracy only amounts to 3 or 4 more texts being correctly classified by classifier #4. Given the small size of the development set, it is therefore difficult to determine whether the performance of classifier 4 is in fact significantly better than the others.

Figure 5.14: Clustering of more complicated joke #1



Finally, the hybrid algorithm (classifier #5), achieved the second best result with an accuracy of 83.3% when it used the results of classifiers #3 and #4 to decide whether a text was a lexical register joke or not.

From Table 5.10 we see that precision scores are very low, indicating that each of the classifiers generate a high number of false positives.

5.12 Conclusion

This chapter began by providing a more detailed model of the structure of lexical register jokes. The simplest lexical register joke consists of three groups of words: a single word with tone A (singleton set #1), words with neutral tone (set #2) and one or more words with a tone B (set #3) which conflicts with tone A. More complicated lexical register jokes have an identical structure except that set #1 contains more than one word. The chapter then introduced and evaluated classifiers #3 and #4 which, like the classifiers #1 and #2, aim to distinguish simple lexical register jokes from regular text.

Classifiers #3 and #4 are similar to classifier #1. For instance all of these algorithms create identical vector spaces and use the cosine metric or a variation of it to compute distances within that space. Classifier #1, however, searched for the most outlying word in a text and used its average cosine distance with the rest of the words in the text to decide if a text is a lexical register joke or not. In other words it divided a text into only two groups - an outlying word and the rest of the words in a text. Classifier #3, however, makes use of a clustering algorithm which attempts to cluster words into the three sets described by our detailed model. It then measures the distances between these clusters and uses this information to judge whether a text is a lexical register joke or not. In this way classifier #3 aims to perform a more detailed dissection and examination of texts

which more closely aligns with our hypothesis about the structure of lexical register jokes.

Classifier #4 is also closely aligned to the more detailed model of lexical register jokes. It looks in a text for the pair of words which are furthest apart in the vector space and uses that distance to decide if the text is a lexical register joke or not.

In spite of their theoretical superiority, however, classifiers #3 and #4's accuracy scores (75% and 77.5% respectively) were no higher than those yielded by classifiers #1 and #2. In fact the 6 newspaper texts misclassified by classifier #3, in tests on the development set, were also misclassified by classifier #1. As explained in chapter 4, classifier #1 appeared to be hampered by a problem of corpus sparsity - corpus set E was created to try to address this issue but the problem remained - and classifiers #3, #4 and #5 (a hybrid of classifiers #3 and #4), which also build a vector space based on corpus set E, appear to be equally affected by this sparsity problem.

Despite not improving on the task of distinguishing between simple lexical register jokes and regular text, an advantage classifiers #3 and #4 confer over classifiers #1 and #2 is that they can theoretically detect the whole class of lexical register jokes: texts in which 1 or more words communicate tone A and 1 or more words represent tone B. A small set of more complicated lexical register jokes was therefore created and tested on each of the classifiers. Classifier #3 performed no better than classifiers #1 and #2, however, in distinguishing more complicated lexical register jokes. The accuracies of all three of these classifiers were nearly identical when tested on the set of more complicated jokes. On the other hand, classifier #4 performed quite well – its accuracy score was 12% - 16% higher than the other classifiers. Given the small size of this test set, however, it is still an open question whether it provides any real advantage in recognizing more complicated lexical register jokes.

Thus neither of the five classifiers tested up to this point stands out as superior. All five will therefore be evaluated on a test set of simple lexical register jokes and the results of that testing is the subject of the following chapter.

Chapter 6

Testing the classifiers

Up to this point, five classifiers have been implemented and tested on a development set of lexical register jokes and regular text. Classifiers #1, #3, #4 and #5 make use of a vector space in which the position of a word is determined by its frequencies in a set of 25 corpora (corpus set E), and each of these classifiers uses the cosine metric or a variation of it to compute distances within that space. Classifier #1, (introduced in Chapter 3) searches for the most outlying word in a text and uses that word's average cosine distance with the rest of the words in the text to decide if a text is a lexical register joke or not. Classifier #3 (introduced in Chapter 5) makes use of a clustering algorithm which attempts to cluster the words of a text into the three sets described by our model of lexical register jokes (see Chapter 5 for details). It then measures the distances between these clusters to judge whether a text is a lexical register joke or not. Classifier #4 (introduced in Chapter 5) looks for the pair of words in a text that are furthest apart from each other in the vector space and uses that distance to classify a text. Classifier #5 (also introduced in Chapter 5), is a hybrid of classifiers #3 and #4 - i.e. both classifiers #3 and #4 have to agree that a text is a lexical register joke if it is to be classified as such. Classifiers #1, #4, and #5 achieved 77.5% accuracy and classifier #3 yielded an accuracy of 75% in distinguishing lexical register jokes from non-jokes in tests on the development set.

Classifier #2 (introduced in Chapter 4) estimates differences in tone in a significantly different way than the others. It makes use of vastly more data - all of the English web pages indexed by Altavista - to determine how frequently words co-occur on the web. These co-occurrence statistics are then used to estimate how similar words might be in terms of tone. The main assumption underlying this algorithm is that the more frequently words co-occur, the more likely it is that they have a similar kind of tone. Despite this algorithm's significant differences from the other four classifiers, it too achieved an accuracy of 77.5% in one of the tests on the development set. Thus none of the five classifiers stands out as superior and the performance of each will be measured in a final test on unseen data. The details and results of that testing are the subject of this chapter.

6.1 Creating the test set

The test data initially consisted of 17 texts which we intuitively regard as lexical register jokes and 17 'regular' texts. All of the jokes are captions from cartoons that appeared in the New Yorker magazine between the years 1969 and 1978. Over 10,000 cartoons were published in the New Yorker during that period but only 30 of these had captions which we intuitively regarded as simple lexical jokes. 13 of those captions went into the development set (which also contained

lexical register jokes from other sources) and 17 into the test set. (See Chapter 3 for details).

The 17 'regular' texts in the test set are newspaper quotes gathered from the November 29 2010 issues of the Canadian Broadcasting News (CBC) and the Globe and Mail news websites. Starting at the homepage of these websites, articles were searched for quotes i.e. speech appearing in quotation marks. Only one quote per article was selected in an attempt to build a varied sample of texts and to avoid bias that might be introduced by taking all the examples from only a few articles. In order to be selected, a quote had to form a complete sentence. Quotes containing proper nouns were rejected, however, because these types of words are unlikely to appear with much frequency in the corpora. Frequency distributions of proper nouns are therefore likely to be poor estimates of tone and so any newspaper quote containing this type of word was rejected as a candidate for entry into the test set.

A total of 40 newspaper quotes were collected from the news websites and the first 17 quotes with roughly the same length as the lexical register jokes were put into the test set. This resulted in a test set in which the average number of words in the lexical register joke texts is 17.6 and the average number of words in the newspaper texts is 18.1. Lexical register jokes in the development set contained an average of 17 words and regular texts were truncated after the 17th word in an attempt to make both kinds of texts equal in length. (See chapter 3). This procedure was not followed, however, in building the test set. Instead, regular texts had to be complete sentences containing at least 12 but no more than 23 words. Lexical register jokes are complete sentences and it was felt that examples of regular text should also be complete sentences so that the lexical register jokes and the regular texts are as structurally and stylistically similar as possible.

6.2 Validating the test set

Before using the test set to evaluate the classifiers, we wanted to ensure that:

1. other people regard the lexical register jokes in the test set as humorous
2. other people do not regard the regular texts in the test set as humorous
3. a conflict of tone between individual words occurs in each of the lexical register jokes

To determine items 1 & 2 above, 20 volunteers were asked to rate texts in the test set. Volunteers' answers were accepted, however, only if a Cloze fluency test was passed. Upon completing the fluency test, volunteers were then asked to score how humorous texts in the test set are on a scale from 0 to 100. Each volunteer's median score was then computed. Any score for a text which exceeded the volunteer's median was regarded as a vote by that volunteer that the text is funny. If at least 65% of the participants voted that a lexical register joke text was humorous then it remained in the test set. Using this procedure, the following four jokes were eliminated from the test set:

- *What it boils down to sire is that they seek a lifestyle more similar to your own.*
- *Attention, everyone! Here comes Poppa, and we're going to drive dull care away! Its quips and cranks and wanton wiles, nods and becks and wreathed smiles.*
- *I wrote this especially for your birthday Pop. It's about how enlightened American business executives have created a viable economy and a mobile and progressive social structure.*

Table 6.1: The test set of lexical register jokes

1	This cell block is for naughty businessmen like yourself who were caught price fixing and such.
2	Hail Caesar! The enemy has been totalled.
3	The woman who ordered the poulet chasseur would like to come in and rap about it.
4	If you do not mind my saying so Doctor, 'Exegetical Dissertations on Theosophical Redirections in the Twentieth Century' will not have quite the impact of 'Keep the Faith Baby!'.
5	Woe! The gods have now decreed that there will be a short pause for a commercial.
6	And finally Lord may thy wisdom guide and inspire the deliberations of those thy servants our planning and zoning commission.
7	I deem thee newsworthy.
8	You are getting to be a big boy sonny and when we are alone it is still nice to hear you call me Popsy. But in front of people try to remember to call me Sire.
9	Miss would you type this searing blast at the honkie power structure in triplicate for me please?
10	There is a skillful juxtaposition of contrasting elements but doggone it I miss his oblique offhand symbolism.
11	I warned you that giving him the presidency would be a boo boo.

- *Oh drat! I forgot to add sodium propionate to retard spoilage.*

Similarly, three newspaper texts were eliminated from the test set because 65% of the respondents regarded them as relatively humorous. These were:

- *He has put a knife in our back, taken us to court and accused us of theft.*
- *If that's art then I'm in the wrong racket. I guess I'm not cultured.*
- *So what that means is I will have to do this every two years until he dies or I die.*

Therefore 13 lexical register jokes and 14 regular texts remained in the test set after this evaluation. Five computational linguists were then asked whether the remaining jokes in the test set possess a conflict of tone which is expressed lexically. Specifically, experts were asked whether a text has a conflict of tone and which words (but not phrases) contribute to that conflict. If at least 3/5 of the linguists agreed that a text contained such a conflict, that text remained in the test set. This evaluation resulted in the elimination of two more jokes. The final test set of lexical register jokes can be seen in Table 6.1

At this point the test set consisted of 11 lexical register jokes and 14 regular texts. To create a balanced set of jokes and regular texts, 3 of the 14 newspaper quotes were therefore randomly eliminated from the test set.

6.3 Results

Table 6.2 shows how the five algorithms performed on the test set. Classifier #3, the clustering classifier, achieved the best test results (shown in bold in the table) with an accuracy of 95.5%,

Table 6.2: Results of the 5 classifiers on the test set (February 2012)

classifier	joke boundary	accuracy	precision	recall	F score
#1 (avg cos)	<= 0.6	19/22 (86.4%)	10/12 (83.3%)	10/11 (90.9%)	86.9
#2 (PMI)	<= -20.8	18/22 (81.8%)	7/7 (100%)	7/11 (63.6%)	77.8
#3 (clustering)	**< = 0.5**	**21/22 (95.5%)**	**10/11 (90.9%)**	**10/11 (90.9%)**	**90.9**
#4 (lowest pairwise cos)	<= 0.25	20/22 (90.9%)	9/9 (100%)	9/11 (81.8%)	90
#5 (hybrid)	<= 0.5 for clustering, <= 0.25 for lowest pairwise cos	20/22 (90.9%)	9/9 (100%)	9/11 (81.8%)	90

90.9% precision and 90.9% recall and classifiers #4 and #5 achieved similarly high results. In tests on the development set, it was determined that the best joke boundary to use for classifier #4 was 0.25, although a boundary of 0.4 yielded only slightly lower results in those earlier tests. Had a boundary of 0.4 been used when classifying the test set, classifier #4 would have achieved exactly the same results as classifier #3.

Classifier #1 also achieved quite good results when evaluated on the test set: 86.4% accuracy, 83.3% precision and 90.9% recall. Classifier #2, which implements the PMI method, yielded the poorest results, although its accuracy of 81.8% and 100% precision are respectably high. This classifier's recall score of 63.6%, however, was significantly lower than those of the other classifiers.

Thus all five of the classifiers performed quite well when evaluated on this test set of unseen data.

6.3.1 Thresholds

Note that the joke boundaries for classifiers #1, #3, #4 and #5 which are shown in Table 6.2 are the boundaries which yielded the highest accuracies in tests on the development set. The boundary for classifier #2, however, was determined by performing an up to date test of the PMI method on the development set. Recall that the PMI method uses co-occurrence counts of words that appear near each other in documents on the web to estimate their similarity in tone. Content on the internet continually changes and grows, however, making co-occurrence statistics change over time. This in turn alters the value of the threshold which the PMI method should use to separate regular text from lexical register jokes. Thus the best threshold to use for the current version of the web was determined using the known set of development texts and this new threshold of -20.8 was used to classify the unknown set of texts in the test set.

Interestingly, this new PMI threshold is approximately 4 times lower than the threshold of -5 which was best to use in tests of the PMI method on the development set 2 years earlier (February 2010). This suggests that Altavista, the web browser used to gather co-occurrence statistics, may have changed its search algorithm somehow and/or content on the internet has significantly altered

in the last two years. Whatever the case may be, the dramatic change in the value of the threshold in the course of 2 years suggests that the data used by classifier #2 may be quite different than its previous incarnation. Despite this possibly dramatic difference, however, the classifier's accuracy score on the test set was only 4% higher in this more recent test. Precision rose from 76.2% to 100%, however, while the recall score fell from 80% to 63.6% in the more recent test.

6.4 Testing with simpler classifiers

In tests on the development set, classifier results were compared to a baseline score that would result if all texts were classified as lexical register jokes. For balanced sets, this naive algorithm will always achieve 50% accuracy, 50% precision and a recall of 100%, resulting in an F score of 66.7. Three additional 'simple' classifiers were implemented, however, in an attempt to assess more thoroughly the performance of the five classifiers on the test set. It is possible that the five classifiers developed so far are overly complicated and that simpler solutions might produce similar (or even better) results. Whatever the likelihood of this may be, it is nonetheless useful to produce various benchmarks against which the five classifiers might be measured.

The four simple classifiers are:

1. a classifier that classifies every text as a lexical register joke.

2. a classifier that classifies a text as a lexical register joke if one or more of the words in the text has a BNC frequency of 0. i.e. it is a rare word. In lexical register jokes, one or more of the words involved in a conflict of tone is often rare. An algorithm that detects rare words and classifies texts containing these types of words as lexical register jokes might perform well.

3. a classifier that uses the same pre-processing, vector space and distance measurements as the non-naive classifier #1 but the initial frequency vector computed for each word is binary: if a word appears in a given corpus, regardless of how often, a value of 1 is assigned to a dimension. If a word does not appear in the corpus, a value of 0 is assigned. Building vectors which simply indicate whether or not a word appears in the various corpora and measuring distances between these vectors might suffice for successful classification.

4. the same classifier as the one described above except that the log entropy processing step is omitted. Recall that the non-naive classifiers, except for classifier #2, compute the frequencies of words in the various corpora and store that data in a matrix where the value of the cell in row i and column j is the normalized frequency count of word i in corpus j. Columns are then weighted according to how much frequencies in that column vary: more variability means a column receives a higher weight. The simple classifier #3 described above, however, creates binary vectors and so the columns of the matrix will be a series of 0s and 1s. When there is less variability in the columns, the pre-processing step taken by the classifier might therefore be unnecessary or may even have a detrimental effect on the classifier.

Table 6.3 shows the performance of more naive classifiers when evaluated on the test set. The highest accuracy achieved by the simpler classifiers was 72.7% (both naive classifiers #2 and #4 received this score). This accuracy is quite high, especially for such a simple algorithm as

Table 6.3: Results of simpler classifiers on test set

classifier	joke boundary	accuracy	precision	recall	F score
classify everything as a joke	n/a	11/22 (50%)	11/22 (50%)	11/11 (100%)	66.7
if text has word with bnc = 0	n/a	16/22 (72.7%)	5/5 (100%)	5/11 (45.5%)	62.5
binary vectors, avg cos method (log entropy)	<= 0.25	13/22 (59.1%)	10/18 (55.6%)	10/11 (90.9%)	69
binary vectors, avg cos method (no log entropy)	**<= 0.7**	**16/22 (72.7%)**	**10/15 (66.7%)**	**10/11 (90.9%)**	**76.9**

classifier #2, and the success of this classifier in particular suggests that the use of rare words in lexical register jokes may be quite common. Nonetheless, the 72.7% accuracy achieved by the best of these naive classifiers is 22.8% lower than the best accuracy achieved by the non-naive classifiers. This suggests that separating lexical register jokes from regular texts may not be a trivial task which can be accomplished with much exactness by algorithms that are simpler than the ones proposed in this thesis.

6.5 Another set of newspaper quotes

Test results of the five classifiers seemed quite high and we wondered whether this might be because the test set of regular texts, which consisted of newspaper quotes, are uncommonly bland. If these texts contain only very common words then distinguishing them from lexical register jokes may be misleadingly simple. Tests were therefore performed on 11 more unvalidated quotes taken from the February 8, 2012 paper version of the Canadian newspaper the Globe and Mail.

Table 6.4 shows the results of this second test and scores are again quite high. Classifiers #4 and #5 achieved the best results and these were exactly the same as the classifier #4 results in the earlier test: 90.9% accuracy, 100% precision and 81.8% recall. Although it did not achieve the best results overall in this new test, classifier #3, still performed well with 90.9% accuracy, 83.3% precision and 90.9% recall. The PMI method, however, performed poorly relative to the other classifiers in this test. It also performed poorly in comparison to its performance on the first test set: accuracy dropped from 81.8% to 72.7%, precision fell by roughly 20% to 77.8% and recall remained at 63.6%.

This second test of the classifiers therefore suggests that the first test may not have been misleadingly positive. It may be, however, that as a genre, newspaper quotes are relatively easy to distinguish from lexical register jokes. Therefore other kinds of regular text were collected and used to evaluate the classifiers and the next section describes the results of this testing.

6.6 Testing with other kinds of regular text

Three additional sets of regular text were created in order to determine how well the five classifiers distinguish these texts from the test set of 11 lexical register jokes. These three additional sets consisted of:

1. 11 randomly selected proverbs

Table 6.4: Running classifiers on unvalidated set of 11 more newspaper quotes

classifier	joke boundary	accuracy	precision	recall	F score
#1 (avg cos)	<= 0.6	19/22 (86.4%)	10/12 (83.3%)	10/11 (90.9%)	86.9
#2 (PMI)	<= -20.8	16/22 (72.7%)	7/9 (77.8%)	7/11 (63.6%)	70.0
#3 (clustering)	< = 0.5	20/22 (90.9%)	10/12 (83.3%)	10/11 (90.9%)	86.9
#4 (lowest pairwise cos)	**<= 0.25**	**20/22 (90.9%)**	**9/9 (100%)**	**9/11 (81.8%)**	**90**
#5 (hybrid)	0.5 for clustering, 0.25 for lowest pairwise cos	20/22 (90.9%)	9/9 (100%)	9/11 (81.8%)	90

Table 6.5: When regular texts are modern proverbs

classifier	joke boundary	accuracy	precision	recall	F score
#1 (avg cos)	<= 0.6	19/22 (86.4%)	10/12 (83.3%)	10/11 (90.9%)	86.9
#2 (PMI)	<= -20.8	16/22 (72.7%)	7/9 (77.8%)	7/11 (63.6%)	70.0
#3 (clustering)	< = 0.5	**20/22 (90.9%)**	**10/11 (90.9%)**	**10/11 (90.9%)**	**90.9**
#4 (lowest pairwise cos)	<= 0.25	19/22 (86.4%)	9/10 (90%)	9/11 (81.8%)	85.7
#5 (hybrid)	0.5 for clustering, 0.25 for lowest pairwise cos	20/22 (90.9%)	10/11 (90.9%)	10/11 (90.9%)	90.9

2. 11 randomly selected New Yorker captions

3. 491 proverbs

Test results are below.

6.6.1 Proverbs

The first 11 proverbs listed on the website http://www.phrases.org.uk/meanings/proverbs.html which contain at least 12 words were collected to create an additional test set of regular texts. Proverbs containing proper nouns or hyphenated words were excluded from the set, however, because frequencies of these types of words in the corpora would probably be misleadingly low.

Table 6.5 shows the results when the five classifiers were given the task of distinguishing these 11 proverbs from the test set of 11 lexical register jokes. Classifiers #3 and #5 scored best of all with 90.9% accuracy, 90.9% precision and 90.9% recall, and these results are very similar to the ones achieved by classifiers #3, #4, and #5 in previous tests in which the set of regular texts consisted of newspaper quotes. That the scores are so similarly high might suggest that proverbs and newspaper quotes possess a similar uniformity of tone which makes them relatively easy to distinguish from the tonal variation that occurs in lexical register jokes[1].

[1]As was the case in the first test described in section 6.3, if classifier #4 used a joke boundary of 0.4 rather than 0.25, scores would be even higher: accuracy would be 95.5%, precision 91.7% and recall 100%.

Table 6.6: When regular texts are New Yorker captions

classifier	joke boundary	accuracy	precision	recall	F score
#1 (avg cos)	<= 0.6	19/22 (86.4%)	10/12 (83.3%)	10/11 (90.9%)	86.9
#2 (PMI)	<= -20.8	16/22 (72.7%)	7/9 (77.8%)	7/11 (63.6%)	70.0
#3 (clustering)	< = 0.5	19/22 (86.4%)	10/12 (83.3%)	10/11 (90.9%)	86.9
#4 (lowest pairwise cos)	**<= 0.25**	**19/22 (86.4%)**	**9/10 (90%)**	**9/11 (81.8%)**	**85.7**
#5 (hybrid)	0.5 for clustering, 0.25 for lowest pairwise cos	19/22 (86.4%)	9/10 (90%)	9/11 (81.8%)	85.7

6.6.2 New Yorker captions

Table 6.6 shows the results when the set of regular texts consists of 11 New Yorker captions that are not lexical register jokes. These texts were randomly selected from a database of New Yorker captions that appeared in the magazine in the year 2003. Captions containing a proper noun or hyphenated words were not admitted into the test set.

Classifiers #4 and #5 produced the best results with 86.4% accuracy, 90% precision and 81.8% recall. Scores are not as high as when the regular text consists of proverbs rather than captions, but results are still quite good.

6.6.3 Large set of proverbs

The classifiers were also tested on a test set consisting of the 11 lexical register jokes and 491 proverbs taken from the site http://www.phrases.org.uk/meanings/proverbs.html. This was the first website to appear in the search results when the term 'proverbs' was searched for on the Canadian version of Google (www.google.ca) in February 2012. Proverbs on the website which contained proper nouns or hyphenated words were not admitted into the set. Results of the test can be seen in Table 6.7.

Classifier #2, which makes use of the Altavista website to estimate the similarity of words in terms of their tone, was not tested on this large test set. Previous test sets had been quite small and automated use of the Altavista search engine had therefore been unobtrusive. Classifying over 500 texts with classifier #2, however, would require making over 10,000 unsanctioned queries to Altavista - not quite a denial of service attack, but the makings of one. Classifier #2 performed worse than the other classifiers in each of the tests performed so far and so its exclusion from this final test is perhaps not a great loss.

From Table 6.7 we see that, as in many of the other tests, classifier #4 performed best overall. It mistakenly classified 17 of the proverbs as lexical register jokes, however, mostly because these proverbs contain words which appear only rarely in the corpora or not at all: words such as 'nowt', 'weepers', 'agley', 'parsnips', 'glisters', 'catchee', 'mickle' and 'muckle'.

6.7 Assessing the test results

All five of the classifiers performed quite well in most of the final tests on unseen data. Classifier #3 yielded the best accuracy scores overall - it yielded an average accuracy of 90.9% in the tests on balanced sets of data - and classifier #4 performed nearly as well with an average of 88.7%

Table 6.7: Testing where regular text consists of 491 modern proverbs

classifier	joke boundary	accuracy	precision	recall	F score
#1 (avg cos)	<= 0.6	434/502 (86.5%)	10/77 (13.0%)	10/11 (90.9%)	22.7
#2 (PMI)	n/a	n/a	n/a	n/a	n/a
#3 (clustering)	< = 0.5	446/502 (88.8%)	10/65 (15.4%)	10/11 (90.9%)	26.3
#4 (lowest pairwise cos)	**<= 0.25**	**483/502 (96.2%)**	**9/26 (34.6%)**	**9/11 (81.8%)**	48.6
#5 (hybrid)	0.5 for clustering, 0.25 for lowest pairwise cos	483/502 (96.2%)	9/26 (34.6%)	9/11 (81.8%)	48.6

Table 6.8: Accuracy scores of the top two performers on tests of the balanced sets data

classifier	test #1	test #2	test #3	test #4	average
#3 (clustering)	95.5%	90.9%	90.9%	86.4%	90.9%
#4 (lowest pairwise cos)	90.9%	90.9%	86.4%	86.4%	88.7%

accuracy. (See Table 6.8 for a summary of these results). Classifier #3 also yielded the highest recall scores overall. In each of the tests on the balanced sets, classifier #3 correctly identified 10/11 (90.9%) of the lexical register jokes whereas classifier #3 correctly identified 9/11 (81.8%) of the lexical register jokes - just 1 joke less than classifier #4 in each of these tests. (See Table 6.9 for details).

Classifier #4, however, yielded the highest precision scores in those same tests. It achieved an average of 95% precision and classifier #3, the second highest performer in this regard, yielded 87.1% precision overall. (See Table 6.10 for a summary of these results).

A test on an unbalanced set of texts, which consisted of distinguishing 491 proverbs from 11 lexical register jokes, yielded high accuracy scores ranging from 86.5% to 96.2% and high recall scores ranging from 81.8% to 90.9%. Precision scores in this test, however, were universally low. The highest precision was a score of only 34.6%, which was achieved by classifier #4.

Such low precision scores in this test have important implications for our plan to adapt the

Table 6.9: Recall scores of the top two performers on tests of the balanced sets data

classifier	test #1	test #2	test #3	test #4	average
#3 (clustering)	10/11 (90.9%)	10/11 (90.9%)	10/11 (90.9%)	10/11 (90.9%)	10/11 (90.9%)
#4 (lowest pairwise cos)	9/11 (81.8%)	9/11 (81.8%)	9/11 (81.8%)	9/11 (81.8%)	9/11 (81.8%)

Table 6.10: Precision scores of the top two performers on tests of the balanced sets data

classifier	test #1	test #2	test #3	test #4	average
#3 (clustering)	90.9%	83.3%	90.9%	83.3%	87.1%
#4 (lowest pairwise cos)	100%	100%	90%	90%	95%

best of the classifiers into a joke generator. Low precision scores yielded by the classifiers in the test of the unbalanced set suggest that a generator based on any one of the classifiers is likely to produce a significant number of texts which are not in fact lexical register jokes. The full algorithm of the generator, which is described in the following chapter, will implement certain filtering however, which will attempt to address the problem of false positives. For instance the generator will have the option to output only texts about which it is fairly confident. Classifiers traditionally do not have this option: by convention they are not allowed to abstain from classifying certain data and must classify all of their input. For this reason filtering was not incorporated into the classifiers.

Because it yielded the highest precision scores (and nearly as high accuracy scores as classifier #3), classifier #4 will form the basis of the lexical register joke generator.

6.8 Conclusion

Test results suggest that the method employed by classifiers #1, #3, #4 and #5 can be quite effective in detecting lexical register jokes. This method consists of building vectors, whose dimensions consist of frequencies in various corpora, and using the distances between these vectors to estimate differences in lexical tone.

Classifier #3 for example, achieved an average accuracy of 90.9%, 87.1% precision, and 90.9% recall in tests of balanced sets of data. Earlier tests on development data suggest that amassing more corpus data might yield even better results, as corpus sparsity seems to be the major limitation of these implementations.

Classifier #2, which uses a different algorithm and considerably more corpus data to estimate differences in tone, did not perform as well as the other classifiers in the series of final tests presented in this chapter. This classifier is not hampered by a shortage of corpus data - it makes use of all the English web pages indexed by Altavista to compute co-occurrence statistics of words and these statistics are used to estimate differences of tone. In spite of this advantage, test results for classifier #2 were consistently worse than the others.

Classifier #4 yielded the highest precision scores of all the classifiers (and ranked a close second after classifier #3 in terms of accuracy and recall scores), and for this reason, it will form the basis of a lexical register joke generator. However in a test on unbalanced data (the last test described in this chapter) all the vector space classifiers, including classifier #4, yielded a high number of false positives (i.e. regular texts mistakenly classified as lexical register jokes). Universally poor precision scores resulted and this has worrying implications for basing a joke generator on any of the classifiers. Filtering mechanisms which attempt to minimize this problem will be implemented into the joke generator, however. These filtering mechanisms, and the full algorithm of the generator, will be described in the following chapter.

Chapter 7

Generator development

7.1 Building a joke generator

Up to this point, algorithms which attempt to discover differences in the tone of individual words have been developed. These algorithms have been incorporated into classifiers which aim to distinguish lexical register jokes from different kinds of regular text. One of the most successful of these was classifier #4 which classifies a text by

- taking words of a text, including stopwords, and computing their frequencies in a set of 25 corpora (corpus set E). A 25 dimensional vector, therefore, is computed for each word and this represents the word's position in a vector space.
- normalizing and performing log entropy transformations of the word vectors.
- using the cosine metric to compute distances between the word vectors.
- finding the pair of words that are furthest apart from each other.
- using that distance to decide whether a text is a lexical register joke or not. If the distance exceeds an empirically determined joke boundary, the classifier concludes that the text is a lexical register joke.

We were interested in determining whether classifier #4 could be modified or incorporated into a system that generates lexical register jokes. In such a system, words that are significantly far apart in the classifier's vector space could be used to create lexical register jokes. An implementation of this idea was therefore developed and the list below outlines the steps it takes to create lexical register jokes:

1. select potential seed texts.
2. eliminate the text if it contains words with extremely sparse frequency information. (See Chapter 4 for details).
3. check the uniformity of tone of the seed text.
4. label parts of speech.
5. select a word in the text.

6. find synonyms for the selected word.

7. for each synonym, determine whether it is a word with extremely sparse frequency information. If so, reject it.

8. build new texts in which synonyms that were not filtered out in step #7 replace the original word.

9. evaluate each new text using classifier #4. If a text exceeds the joke boundary, it is considered a lexical register joke.

The rest of the chapter will discuss each of these steps in turn.

7.2 Step 1: select potential seed texts

Lexical register jokes do not simply consist of words with conflicting tone. If this were the case, a simple list of words with incongruous tone such as "poulet, woman, come, chasseur, ordered, rap" would be humorous.

Even though this list consists of the content words of a lexical register joke (which a majority of human subjects classified as humorous - see chapter 6 for details), and contains at least one pair of words with conflicting tone (in this case there are 2 pairs of words in conflict – 'rap' and 'poulet' and 'rap' and 'chasseur'), the humour of the joke dissolves when the text is converted into a simple list of words. This example suggests therefore that the context in which an incongruity of tone appears matters. At the very least, an incongruity of tone probably needs to appear within a semantically coherent phrase or sentence.

Automatic generation of semantically coherent sentences is a difficult problem, however, which the field of natural language generation has yet to solve. To avoid having to generate a coherent sentence from scratch, therefore, the lexical register joke generator will instead take an already existing text, assume that it is coherent, choose one of the words in this text and attempt to find a synonym for it whose tone conflicts with the tone of one or more other words in the text. Simply replacing a word with a near-synonym is unlikely to render the text incomprehensible, and having done so will have introduced a potentially humorous incongruity of tone into the text.

Creating an incongruity of tone within just any kind of coherent sentence might only occasionally produce a lexical register joke, however. This is because lexical register jokes probably also need to possess the property of self-containment. A lexical register joke is read and understood on its own without supporting text and seed texts that are to be turned into lexical register jokes would also need to possess this property. Equipping a system to recognize whether a text makes sense on its own and can somehow stand alone is currently impossible however and may be an AI-complete problem. An alternative to having to rely on such a system is to identify by hand certain genres or types of writing which are likely to contain self-contained passages and to provide these as input to the lexical register joke generator. The generator could then randomly pick passages belonging to these genres and use them as potential seed texts for lexical register jokes.

Two types of writing which are likely to contain self-contained passages are proverbs and quotations. Proverbs are often quoted in conversation or text to summarize a particular idea and so

the context in which they appear is important. Yet they also have a kind of distance or separateness to them - they frequently stand apart from the conversation or text in which they appear - and quotations often display this same kind of detachment.

Therefore biblical proverbs and quotations attributed to famous figures will, in separate runs, act as input to the generator. These types of texts are self-contained and they display a formality of tone which the generator can attempt to oppose in a humorous way by making certain lexical changes. Other kinds of self-contained seed texts could also be used of course, but in the initial tests of the generator, examples from these two types of genres will be used.

7.3 Step 2: eliminate unsuitable seed texts

In this step, the system will eliminate certain texts from the set of seed texts. These texts will be eliminated because they contain problematic words whose tone is difficult to estimate. In Chapter 4, profile A words were loosely defined as words with low frequencies in only a few corpora and words with profile C were defined as words that do not appear in any corpora. Words with these profiles were problematic for the vector space classifiers because at least two types of words, which are quite different and should be distinguished from each other, can have these kinds of profiles: (1) truly rare words (such as 'sesquipedalianism' for example) (2) words which are not actually rare but only appear to be so because the corpus data used by the system is simply too sparse. The latter words have misleadingly low frequency distributions (i.e. have profiles A or C) and will incorrectly appear as outliers in the vector space, thus making the texts containing them incorrectly appear to possess an incongruity of tone. For this reason, the generator will reject seed texts containing profile A or C words.

In the initial test of the generator, a word with low frequency in a corpus will be defined as a word that appears fewer than 4 times per million in that corpus - call this the **frequency threshold -** and a word that appears in "only a few corpora" will be defined as a word that appears in fewer than 2 corpora - call this the **corpus threshold**. In other words, a profile A word will initially be defined as a word that appears with a frequency of 4 or more times per million in only 1 corpus. Different values for the corpus and frequency thresholds will be tried in later tests of the generator.

7.4 Step 3: check uniformity of tone

The next step in the generation algorithm attempts to evaluate the uniformity of tone in a potential seed text. Seed texts containing words that are significantly far apart from each other are considered suspect and are eliminated in this step of the algorithm. Although seed texts containing profile A or C words will have been filtered out in step 2 above, some of the texts passing through that filter might nonetheless contain words with inaccurate frequency profiles. As explained in Chapter 4, the corpora used to compute frequency vectors of words do not always provide adequate coverage. Consequently, words will often appear further away from each other than they should be in the vector space and the classifiers (described in chapter 5) are led into making false positive errors. (Recall that false positives are regular texts that are misclassified as lexical register jokes).

For instance in tests on an unbalanced set of texts (consisting of 11 lexical register jokes and 491 proverbs - see Chapter 6 for details) classifier #4, the algorithm that will be incorporated into the generator, produced the fewest number of false positives of all the classifiers, but the number

of these was nonetheless quite high. 7 of the 491 proverbs were incorrectly identified as lexical register jokes while 9 of the 11 lexical register jokes were classified correctly. This resulted in a low precision score of 34.6%.

$$precision = \frac{tp}{\|selected\|} = \frac{tp}{tp+fp} = \frac{9}{9+7} = 34.6\%$$

The system therefore seems prone to making false positives. In light of this, a seed text - i.e. a regular text which typically has uniform tone - which is found to contain words that are significantly far apart in the space, will be regarded with suspicion.

It is important to note that sometimes the generator will be correct in its assessment that a regular text is on the verge of containing an incongruity of tone - in which case replacing a word with slightly different tone might suffice in transforming the text into a lexical register joke. It seems likely that texts in the real world have varying degrees of uniformity of tone and that some regular texts are on the verge of containing an incongruity of tone. In other words, if regular texts were plotted in a vector space that perfectly represented their tone, it is possible that for some, the distances between words would be quite large. As we have seen, however, the vector space we have developed is not perfect and has trouble differentiating these true cases from cases arising from corpus sparsity problems. Given these problems, the generator will not attempt to construct lexical register jokes using seed texts that appear to already contain or nearly contain an incongruity of tone.

In developing classifier #4, it was determined that a cosine value of 0.25 was the best joke boundary to use to decide whether a text is a lexical register joke; call this the **joke boundary**. In other words, any text containing a pair of words whose distance exceeded this value was classified as a lexical register joke. This stage of the algorithm, however, aims to filter out texts that are not significantly uniform in tone. We therefore make use of a value - call this the **uniformity boundary** - which texts must exceed in order to be considered uniform. More specifically, the cosine of each pair of words in a text will need to exceed this boundary in order for the text to be considered uniform.

Values for the joke boundary and the uniformity boundary will be varied in the development testing of the joke generator, but not independently of each other because the uniformity boundary will always need to be higher than the joke boundary. As Tables 7.1 - 7.2 show, values up to 0.55 could have acted as effective joke boundaries for separating lexical register jokes from regular text (in the testing of classifier #4 on a series of final test sets, described in the previous chapter). Using a joke boundary as high as 0.55 for instance would have yielded 90.9% accuracy, 84.6% precision and 100% recall in separating lexical register jokes from regular text (in this first of the final set of tests). These tests of the classifiers suggest therefore that a lower bound for the uniformity boundary should probably not be much lower than 0.55. In initial tests of the generator, a text which is considered uniform in lexical tone will therefore be defined as one in which the cosine values between each pair of words in the text are all greater than 0.6, but different values for this parameter will be tried in later tests of the generator. (Note that the generator therefore makes use of four parameters: a corpus threshold (see Section 7.4 for details), frequency threshold (see Section 7.4), uniformity boundary and joke boundary).

Table 7.1: Classifier #4 results on test set #1 (lexical register jokes and newspaper quotes) using different thresholds

joke boundary	accuracy	precision	recall	F score
0.25	20/22 (90.9%)	9/9 (100%)	9/11 (81.8%)	90.0
0.3	20/22 (90.9%)	9/9 (100%)	9/11 (81.8%)	90.0
0.35	20/22 (90.9%)	9/9 (100%)	9/11 (81.8%)	90.0
0.4	22/22 (100%)	11/11 (100%)	11/11 (100%)	100
0.45	21/22 (95.5%)	11/12 (91.7%)	11/11 (100%)	95.7
0.5	20/22 (90.9%)	11/13 (84.6%)	11/11 (100%)	91.7
0.55	20/22 (90.9%)	11/13 (84.6%)	11/11 (100%)	91.7
0.6	15/22 (68.2%)	11/18 (61.1%)	11/11 (100%)	75.9
0.65	14/22 (63.6%)	11/19 (57.9%)	11/11 (100%)	73.3

Table 7.2: Classifier #4 results on test set #2 (lexical register jokes and newspaper quotes#2) using different thresholds

joke boundary	accuracy	precision	recall	F score
0.25	20/22 (90.9%)	9/9 (100%)	9/11 (81.8%)	90.0
0.3	19/22 (86.4%)	9/10 (90%)	9/11 (81.8%)	85.7
0.35	19/22 (86.4%)	9/10 (90%)	9/11 (81.8%)	85.7
0.4	21/22 (95.5%)	11/12 (91.7%)	11/11 (100%)	95.7
0.45	20/22 (90.9%)	11/13 (84.6%)	11/11 (100%)	91.7
0.5	20/22 (90.9%)	11/13 (84.6%)	11/11 (100%)	91.7
0.55	18/22 (81.8%)	11/15 (73.3%)	11/11 (100%)	84.6
0.6	18/22 (81.8%)	11/15 (73.3%)	11/11 (100%)	84.6
0.65	15/22 (68.2%)	11/18 (61.1%)	11/11 (100%)	75.9

7.5 Step 4: label parts of speech

The generator will then label the parts of speech in a seed text using the Stanford parser (Klein and Manning, 2003). This parser uses a probabilistic context-free grammar to identify the syntactic parts of a text.

7.6 Step 5: select a word in the text

The generator makes a single change to a seed text and that is to replace a word with a synonym whose tone conflicts with one or more other words in the text. Different parts of speech, for instance verbs or adjectives, could be selected but in our tests we have arbitrarily chosen nouns as targets for replacement. The noun nearest the end of a text is selected so that a certain kind of tone can first of all be established and then undermined.

7.7 Step 6: find synonyms for the selected word

The generator then automatically retrieves synonyms for the targeted word from the website www.thesaurus.com. This website has been selected because it provides synonyms whose tones seem to vary in their tone (whereas a resource such as WordNet does not appear to provide as much variability of tone). For example, some of the synonyms suggested for the word 'happy' are "blessed, blissful, blithe, chipper, chirpy, convivial, delighted, ecstatic, jolly, jubilant, peppy, perky, sparkling, tickled" and these seem to demonstrate a wide variation of tone.

Sometimes a synonym suggested by thesaurus.com will be a phrase. For example one of the synonyms of 'child' listed on the website is "little one". We arbitrarily allowed the generator to accept phrasal synonyms but it is important to note that the system will not evaluate the tone of the phrase as a whole because it can only compute frequencies for unigrams. Thus in our example, a frequency vector would not be created for the noun phrase "little one" - instead a vector for the word 'little' and another vector for the word 'one' would be created and plotted into the space along with the other unigrams in the text.

Note that the generator does not perform word sense disambiguation. In the text "As our case is new, we must think and act anew.", for example, the last occurring noun is the word 'case' and its meaning here is "case as in situation". When the generator retrieves synonyms of the word 'case' from thesaurus.com, however, the synonyms returned are for the word "case as in container". Thus errors will sometimes arise because synonyms are retrieved for the wrong meaning of a word.

7.8 Step 7: reject synonyms with problematic profiles

The generator checks each synonym to determine whether it is a profile A or C word. If it is, the synonym is rejected because it is considered a problematic word, for the reasons given in Section 7.3.

7.9 Step 8: create a new text

The generator then builds a new text for each synonym which was not filtered out in step 7. In the new text, a synonym appears in place of the word that was selected in Step 5 (see Section 7.6).

Table 7.3: Setting joke boundary to 0.25 and 0.4 in final tests of classifier #4

final test no.	**precision when joke boundary is 0.25**	**precision when joke boundary is 0.4**
#1 lexical register jokes and newspaper quotes	9/9 (100%)	11/11 (100%)
#2 lexical register jokes and another set of newspaper quotes	9/9 (100%)	11/12 (91.7%)
#3 lexical register jokes and modern proverbs	9/10 (90%)	11/12 (91.7%)
#4 lexical register jokes and new yorker captions (which are not lexical register jokes)	9/10 (90%)	11/13 (84.6%)
#5 lexical register jokes and large set of modern proverbs	9/25 (36%)	11/47 (23.4%)

7.10 Step 9: evaluate each new text using classifier #4

Each new text is then evaluated using classifier #4. Section 7.1 outlines the algorithm of this classifier and Chapter 5 provides further details. In development testing of classifier #4, it was determined that a cosine value of 0.25 was the best boundary to use to decide whether a text is a lexical register joke (i.e. the joke boundary was set to 0.25) but as Table 7.1 shows, boundaries up to 0.55 yielded high performance scores in the first of these final tests. Table 7.3 compares the precision scores yielded by classifier #4 on the five final test sets when the joke boundary was set to 0.25 and 0.4 respectively. Precision scores were very similar in each case and so both of these values, among others, will be experimented with in tests of the generator. We are interested in maximizing precision when generating jokes because precision measures the extent to which items identified by the system as lexical register jokes are truly lexical register jokes. In other words reducing the number of false positives is the priority here. Accuracy scores could instead be used to measure performance but computing these would be considerably more onerous. This is because, for each set of seed texts, potentially hundreds of texts can be constructed in step 8 of the algorithm. If a text passes steps 2 and 3 of the algorithm, as many as 30 or more synonyms could be retrieved from thesaurus.com and so each seed text could produce 30 possible lexical register jokes to be evaluated by the system. If, for example, 40 seed texts are input into the generator, 1200 texts could then be created in step 8 (40 seed texts * 30 synonyms per seed text = 1200 generated texts). Computing an accuracy score would require computing the following ratio: (number of true positives + number of true negatives)/total number of texts. In other words all 1200 texts would have to be evaluated by volunteers to determine how many texts the generator was right to generate or reject. Computing a precision score, on the other hand, only requires evaluating whether the texts actually output by the generator are lexical register jokes.

7.11 Development testing of the generator

In development testing, different values for the four thresholds (described above) were experimented with and our intuition was used to decide whether a text output by the generator contains a large difference in tone. Formally judging whether texts are humorous or not was deferred until the final test, when texts output by the generator were evaluated by human volunteers. As explained above, precision scores arising from various threshold settings used in these trials helped determine the threshold values used in the final tests of the generator. (See Chapter 8 for details).

7.11.1 Test #1 - initial values for thresholds

In the first set of development testing, 57 proverbs (chapters 3 and 13 of the book of Proverbs in the King James version of the bible) were input to the generator as potential seed texts. In a second and separate run, 38 quotes attributed to Abraham Lincoln were then provided as input. In these initial tests, the four thresholds of the generator were set as follows:

- uniformity boundary = 0.6 (i.e. all the word pairs in the seed texts had to have cos distances >= 0.6 for the text to be considered uniform in tone).
- joke boundary = 0.25 (i.e. texts that are classified as lexical register jokes contain at least one word pair with a cosine value <= 0.25).
- frequency threshold = 4 (i.e. a word has to have a frequency > 4 for it to be considered as appearing in a corpus)
- corpus threshold = 1 (i.e. a word appearing in just 1 corpus is considered a profile A word)

Using these settings, the texts output by the generator when proverbs were used as potential seed texts were:

> So shall they be life unto thy soul, and grace to thy cervix.
>
> So shall they be life unto thy soul, and grace to thy isthmus.
>
> Strive not with a man without cause, if he have done thee no disservice.
>
> Strive not with a man without cause, if he have done thee no impairment.

The original texts on which these jokes are based are “So shall they be life unto thy soul, and grace to thy neck” and “Strive not with a man without cause, if he have done thee no harm”. Only 2 of the 57 proverbs ended up being used as seed texts for lexical register joking and the other 55 proverbs were rejected for one of the reasons below:

1. the text exceeded the uniformity boundary
2. the text contained problematic profile A or C words
3. synonyms for the word to be replaced were either all problematic or when plotted into the space, did not exceed the joke boundary.

We believe that only the first two texts output by the generator contain an incongruity of tone whereas in the other two texts, the tone of the words ‘disservice’ and ‘impairment’ do not, according to our intuition, create significant clashes of tone in their respective texts.

Table 7.4: Original seed texts (Lincoln quotes) and generator's version

Original Lincoln quote	Generator version
Avoid popularity if you would have peace.	Avoid popularity if you would have amity.
Do not interfere with anything in the Constitution. That must be maintained, for it is the only safeguard of our liberties.	Do not interfere with anything in the Constitution. That must be maintained, for it is the only safeguard of our authorization.
Everybody likes a compliment.	Everybody likes an acclamation.
I am a firm believer in the people. If given the truth, they can be depended upon to meet any national crisis. The great point is to bring them the real facts.	**I am a firm believer in the people. If given the truth, they can be depended upon to meet any national crisis. The great point is to bring them the real dope.**
I can make more generals, but horses cost money.	**I can make more generals, but horses cost pesos.**

Similarly, Table 7.4 shows generator results when quotes attributed to Abraham Lincoln (taken from www.brainyquote.com) are used as seed texts. The first column of this table provides the original quote and the second column shows the version of the text output by the generator. In this test run of the generator, we think that the last two texts, shown in bold in the table, contain an incongruity of tone whereas the others do not.

7.11.2 Test #2 - varying the uniformity and joke boundaries

When the frequency and corpus thresholds are kept at 4 and 1 respectively but the uniformity and joke boundaries are varied (as shown in the leftmost column of Table 7.5), the number of texts output by the generator increases. Table 7.5 shows the proportion of texts output by the generator which we believe contain significant differences in lexical tone when the 57 biblical proverbs are used as seed texts and Table 7.6 shows the results when the 38 Lincoln quotes act as seeds.

Appendix D provides a complete listing of the texts output by the generator at the different settings. Many texts (too many to list here) were mistakenly classified as containing incongruities of tone and were output by the generator. Some examples are:

> Always bear in mind that your own **obstinacy** to succeed is more important than any other (original word was 'resolution').
>
> The law of the wise is a fountain of life, to depart from the snares of **bereavement** (original word was 'death').
>
> Be not wise in thine own eyes: fear the LORD, and depart from **wrongdoing** (original word was 'evil').

The word appearing in bold in a text is the synonym which replaced a word in the text and this synonym was significantly further away from at least one other word in the text. We believe the system made an error in these cases because of the corpus sparsity problem described in chapter 4. Because the set of corpora used by the system does not provide enough coverage, the frequencies for the bolded words are, we think, misleadingly low, and make the words appear incorrectly as outliers in their respective texts. The frequency vectors for the bolded synonyms are:

Table 7.5: Percentage of texts which contain a significant difference in tone when frequency threshold = 4, corpus threshold = 1 and 57 proverbs (KJV) are used as seed texts. The threshold settings in the last row of the table, shown in bold, produce the greatest number of outputs and relatively high precision scores. They will be used in the final test of the generator.

uniformity and joke boundaries	**incongruous texts**	**examples**
0.6 and 0.25	2/4 (50%)	-So shall they be life unto thy soul, and grace to thy cervix. -So shall they be life unto thy soul, and grace to thy isthmus.
0.6 and 0.4	2/8 (25%)	-So shall they be life unto thy soul, and grace to thy cervix. -So shall they be life unto thy soul, and grace to thy isthmus.
0.55 and 0.35	5/22 (22.7%)	-The law of the wise is a fountain of life, to depart from the snares of euthanasia. -The law of the wise is a fountain of life, to depart from the snares of exit.
0.5 and 0.35	**~16/41 (39.0%)**	-Withhold not good from them to whom it is due, when it is in the power of thine shaker to do it. -The wise shall inherit glory: but shame shall be the promotion of blockheads.

Table 7.6: Percentage of texts which contain a significant difference in tone when frequency threshold = 4, corpus threshold = 1 and 38 Lincoln quotes are used as seed texts. The threshold settings in the last row of the table, shown in bold, produce the greatest number of outputs and the highest precision score. They will be used in the final test of the generator.

uniformity and joke boundaries	**incongruous texts**	**examples**
0.6 and 0.25	2/5 (40%)	-I am a firm believer in the people. If given the truth, they can be depended upon to meet any national crisis. The great point is to bring them the real dope. -I can make more generals, but horses cost pesos.
0.6 and 0.4	15/43 (34.9%)	-A condo divided against itself cannot stand. -Am I not destroying my enemies when I make bosom buddies of them?
0.55 and 0.35	14/34 (41.2%)	-I can make more generals, but horses cost loot. -I am a firm believer in the people. If given the truth, they can be depended upon to meet any national crisis. The great point is to bring them the real brass tacks.
0.5 and 0.35	**16/37 (43.2%)**	-A residency divided against itself cannot stand. -I desire so to conduct the affairs of this administration that if at the end... I have lost every other friend on earth, I shall at least have one friend left, and that roommate shall be down inside of me.

Table 7.7: Percentage of Lincoln seed texts which contain a significant difference in tone when frequency threshold = 20, corpus threshold = 2. The threshold settings in the third row of the table, shown in bold, produce the highest precision scores. They will be used in the final test of the generator.

uniformity and joke boundaries	**incongruous texts**	**examples**
0.6 and 0.25	0 texts output	n/a
0.6 and 0.4	3/4 (75%)	-Am I not destroying my enemies when I make bosom buddies of them? -I can make more generals, but horses cost bucks.
0.55 and 0.35	**3/3 (100%)**	-Am I not destroying my enemies when I make bosom buddies of them? -I destroy my enemies when I make them my bosom buddies.
0.5 and 0.35	3/3 (100%)	-Am I not destroying my enemies when I make bosom buddies of them? -I destroy my enemies when I make them my bosom buddies.

obstinacy 0 0 1 0 0 0 0.67 0 0.89 31.5 0 0 0 0 0 0 0 0 0 0.59 1.19 0 5.64 0 0

bereavement 0 0 4 0 0 0.75 0 5.45 0 0 0 0 14.47 0 0 0 0 0 1.18 1.79 0 2.11 7.52 0 0

wrongdoing 0 0 1 1.35 0 2.26 2.02 2.72 0 0 0 1.16 0 0 3.08 0 0 6.93 0 0 4.17 2.11 0 0 0

Each number represents the word's normalized count in a given corpus. We find that these words only barely exceeded the frequency and corpus thresholds and so were not classified as profile A words. (Recall that in this test run, a word had to have a frequency greater than 4.0 for it to be considered as appearing in a corpus and had to appear in more than 1 corpus in order to avoid being classified as a profile A word). These words probably should have been identified as problematic, however, because we believe they mistakenly appear as outliers in the space and incorrectly made the texts containing them exceed the joke boundary. The corpus and frequency thresholds were therefore raised in the test below in an effort to eliminate the number of false positives generated by the system, which we believe arise principally from the corpus sparsity problem.

7.11.3 Test #3 - raising the frequency and corpus thresholds

In another run of development testing, the frequency threshold was raised to 20 and the corpus threshold was raised to 2. Only one setting (uniformity 0.5, joke 0.35) produced any texts (2) and neither of these showed any incongruity.

Table 7.7 shows results when Lincoln quotes are used as seed texts using these threshold settings. More seed texts are rejected in all the tests because the definitions of profile A and C words are wider and so fewer texts were output by the generator.

In fact in many cases no texts whatsoever were output, thus revealing a dilemma. If corpus and frequency thresholds are too low (as they may have been in test #2), too many words with

misleadingly low distributions pass through the filters and are allowed to generate false positives. When corpus and frequency thresholds are raised, however, as they were in this test run (test #3), too many words are classified as problematic profile A words and at most only a handful of texts are output by the generator. It could be that the corpus and frequency thresholds were raised by too much in test #3, but previous testing of classifier #4 and the other vector space classifiers (see chapter 4 for details) suggests that tweaking these thresholds can only accomplish so much. Doing so will not solve the underlying corpus sparsity problem which we believe is the main problem affecting the performance of the vector space classifiers and the generator.

7.12 Parameters for final test

For the final tests of the generator, a set of thresholds to use might be frequency threshold = 20, corpus threshold = 2, uniformity boundary = 0.55 and joke boundary = 0.35 because these settings produce the highest precision scores in development testing. The generated output arising from these settings, however, shows little variation. For example all the jokes output from these settings, when proverbs acted as input to the generator, used the same expression to create a clash of tone:

> Am I not destroying my enemies when I make **bosom buddies** of them?
>
> I desire so to conduct the affairs of this administration that if at the end... I have lost every other friend on earth, I shall at least have one friend left, and that **bosom buddy** shall be down inside of me.
>
> I destroy my enemies when I make them my **bosom buddies**.

The words in the expression "bosom buddy" were the only words that managed to both pass the filtering step (step 7 above) and to create a conflict of tone and so the variety of the output is severely limited at these settings. In fact the pair of words exceeding the joke boundary in texts #1 and #3 turned out to be 'bosom' and 'buddy'. In other words, the two words making up the synonym returned by thesaurus.com generated the incongruity of tone with each other rather than with other words in the text. (In text #2 'bosom' and 'buddy' exceeded the joke boundary as did the word pair 'bosom' and 'administration').

More numerous texts with more interesting variation can be seen in the generator's output when the following settings are used: frequency threshold = 4, corpus threshold = 1, uniformity boundary = 0.5 and joke boundary = 0.35. Precision is 39.0% when seed texts consisted of proverbs and 43.2% precision resulted when Lincoln quotes were used. These settings produce the second highest precision scores and so the generator will use the following two sets of settings in the final testing:

- the more restrictive thresholds (frequency threshold = 20, corpus threshold = 2, uniformity boundary = 0.55 and joke boundary = 0.35) which yield the highest precision scores, but fewer and less interesting results.
- the less restrictive thresholds (frequency threshold = 4, corpus threshold = 1, uniformity boundary = 0.5 and joke boundary = 0.35) which yield lower precision scores but generate more interesting results.

Details and results of the final test of the generator are discussed in the following chapter.

7.13 Summary

A system for generating lexical register jokes was described and developed in this chapter. The pre-processing methods, vector space, and distance metric used by classifier #4, one of the most successful of the classifiers tested in chapter 6, were incorporated into a joke generator. The generator attempts to modify seed texts (i.e. regular texts that are not lexical register jokes) in such a way that at least one pair of words in a text are significantly far apart from each other in a vector space in which position is an estimate of tone. The program makes use of four parameters, described in Sections 7.3 and 7.4, and different values for these parameters were tried in development testing of the system. Two sets of parameters emerged as the most promising and both sets will be used in a final test of the generator. The following chapter discusses the details and results of that final test.

Chapter 8

Evaluation of the generator

The previous chapter described the development of a generator which attempts to construct lexical register jokes from non-humorous seed texts. This chapter describes a final test of the generator and evaluates its performance in that test. Specifically, this chapter explains which potential seed texts were used in the test, how the generated texts were evaluated and how humorous the generated texts were considered in comparison to a set of human-made lexical register jokes and non-humorous texts.

8.1 The potential seed texts

Potential seed texts for the final test consisted of:

- 40 quotes attributed to Barrack Obama (from the website www.brainyquote.com).
- 40 quotes attributed to Richard Nixon (from the website www.brainyquote.com).
- 73 quotes attributed to Donald Rumsfeld (from the website www.brainyquote.com).
- 50 proverbs from chapters 6 and 17 from the Book of Proverbs (from the King James version of the bible).

The two chapters of proverbs were randomly selected from the 31 chapters of the bible's Book of Proverbs. These chapters contain 63 verses, however each verse does not always form a complete proverb or passage which can stand alone. For example verse 1 of chapter 6 does not even form a complete sentence:

> My son, if thou be surety for thy friend, if thou hast stricken thy hand with a stranger,

Verse 2 is required to complete the sentence (and the thought):

> thou art snared with the words of thy mouth, thou art taken with the words of thy mouth.

Verse 3, on the other hand, seems relatively self-contained:

> Do this now, my son, and deliver thyself, when thou art come into the hand of thy friend; go, humble thyself, and make sure thy friend.

We therefore used our intuition in this way to group the 63 verses of chapters 6 and 17 into 50 relatively independent/self-contained passages.

Table 8.1: Two sets of generator parameters were used in the final test

set #1	set #2
• frequency threshold = 20 • corpus threshold = 2 • uniformity boundary = 0.55 • joke boundary = 0.35 These more restrictive parameters yielded the highest precision scores in development testing, but fewer and less varied texts were generated.	• frequency threshold = 4 • corpus threshold = 1 • uniformity boundary = 0.5 • joke boundary = 0.35 These less restrictive parameters yielded lower precision scores in development testing but more texts were generated.

8.2 Results of generation

As explained in chapter 7, the generator uses 4 parameters and different values for these parameters were experimented with in development testing. The two sets of parameters shown in Table 8.1 yielded the best results and both sets were used in the final test.

The more restrictive parameters (set #1 in Table 8.1) generated only 2 outputs from the 203 seed texts:

> If you develop decree, never have more than ten.
>
> Presidential leadership need not always cost money. Look for low and no dues options. They can be surprisingly effective.

The seed texts for these outputs were: "If you develop rules, never have more than ten" and "Presidential leadership need not always cost money. Look for low and no cost options. They can be surprisingly effective." These are quotes attributed to Donald Rumsfeld.

The less restrictive parameters (set #2 in Table 8.1), however, produced considerably more results: 19 generated texts issued from the Obama quotes, 48 texts from the Donald Rumsfeld quotes, 22 from the Nixon quotes and 11 outputs from the proverbs. All 100 of these texts are listed in Appendix D. Given that so few texts were generated using the set #1 parameters, only the texts arising from the second, less restrictive, set of settings were evaluated.

100 generated texts are too many to be evaluated, however, and so further filtering was performed by lowering the joke boundary to 0.25 and adding the restriction that generated output could contain only 20 or fewer words. Lexical register jokes tend to be quite short - the average length of a set of attested lexical register jokes (used in the final test set for classifier testing - see Chapter 6 for details) was 17.6 - and restricting the number of words in this type of humour may be important. This filtering reduced the number of generated texts in the final set from 100 to 27. Call this set of generated texts set A.

Sometimes multiple jokes in set A were generated from the same seed text. For example:

> Can a man take fire in his bosom, and his equipment not be burned?

Can a man take fire in his bosom, and his outfit not be burned?

In one case, as many as 5 texts in set A issued from the same seed text. We decided that in the evaluation a volunteer should judge only 1 text spawned from a given seed text because:

1. the volunteer might tire from having to judge texts that are nearly identical.

2. we do not want the volunteer to unfairly compare nearly identical texts with each other more than they compare those texts with other texts in a questionnaire.

Therefore, in order to avoid having two or more similar texts appear in a volunteer questionnaire, set A was split into 5 groups by dividing the 27 texts into 3 groups of 5 generated texts and 2 groups of 6 generated texts. None of the texts in set A appear in more than 1 group and no two texts issuing from the same seed text appear in the same group.

Control texts were then added to the 5 groups. These 15 control texts consist of:

- five human-made lexical register jokes. These are captions taken from New Yorker cartoons. They were randomly selected from a group of 13 lexical register jokes which, during classifier testing, were attested as humorous by a majority of people. (See Chapter 6 for details).

- five newspaper quotes which were attested as not humorous by a majority of people in classifier testing. (See Chapter 6 for details).

- five unaltered proverbs randomly chosen from the book of Proverbs (King James version). Given that some of the generated texts are based on proverbs, we wanted to determine whether people regard unaltered proverbs as humorous.

The same 15 control texts were put into each of the 5 groups.

8.3 Evaluation of the generated texts

Five different online questionnaires were created from the five sets of texts discussed above. Texts were randomly ordered within each volunteer's questionnaire and the volunteer was asked to score, on a scale from 0 to 100, how humorous the texts are. 150 people answered the questionnaire but a Cloze fluency test (the same one used in classifier testing – see Chapter 6 for details) had to be passed for results to be accepted. Only 56 people passed this test. At least 10 different people evaluated each of the generated texts and because the same control texts appeared in each of the 5 questionnaires, each of the control texts was evaluated by 56 people. An excerpt of the instructions for the questionnaire is shown below:

> We are studying a subtle kind of humor. In this kind of text, words which don't really fit together appear in a text and the result is sometimes funny. For example old and modern words might appear in the same text. Or formal and informal words might appear in the same text. Here's an example:
>
> *Gentlemen! Nothing stands in the way of a final accord except that management wants profit maximization and the union wants more moola.*

> In this example, many of the words of the text are formal but then the informal word moola clashes with these words and the result is humorous. Some of the texts below have this funny kind of clash while other texts do not. Note that the funny texts will not be humorous in the way a comedian's stand up joke is humorous. The funny texts we're studying are a subtle kind of humor which involves mixing together words that somehow conflict or clash with each other so please read the texts carefully with this in mind. On a scale from 0 to 100, please rate the funniness of each of the texts below.

It is possible that a volunteer's score for a text does not rate the actual funniness of the text but only indicates whether he believes there is a conflict of tone in the text. Evalutions with different instructions might be tried in future work. For example the disclaimer that texts being evaluated will likely not be as humorous as a stand-up comedian's jokes might also be edited or removed entirely.

8.4 Analysis of the results

Volunteers judge, on a scale from 0 to 100, how funny they think a text is. People are likely to have different averages and ranges to their scores, however (i.e. volunteer scores will have different distributions). For example let us say that only 5 texts are evaluated and subject #1's scores are [0, 0, 20, 15, 15] and subject #2's scores are [10, 10, 20, 80, 90]. A score of 20 in the first distribution, which only ranges from 0 to 20, is high relative to the other scores in the distribution. On the other hand a score of 20 in the second distribution, which has a wider range of scores from 10 to 90, is relatively low. Although both subjects have given text #3 the same raw score, the different ranges of the distributions reveal important differences in how the two subjects regard the text. Subject #1 considers it more humorous than all the other texts whereas subject #2 regards it as less humorous than some of the other texts that were evaluated. How a text scores in comparison to other texts is important information - we are interested in determining whether people regard the generated texts as more or less humorous than non-jokes and human-made jokes - but this information is lost when only raw scores from different distributions are compared.

To capture where a value stands in relation to other values in a distribution and to make values from different distributions more comparable, scores can be standardized. A standard score (or zscore) expresses a raw score in terms of how many standard deviations it is away from the mean of a given distribution:

$$z = \frac{X - \mu}{\sigma}$$

In the equation, X is the raw score, $\mu$is the mean and σ is the standard deviation. In the example above, a raw score of 20 in the first distribution yields a zscore of 1.07 whereas a raw score of 20 in the second distribution produces a zscore of -0.56. In other words a score of 20 in the first distribution is 1.07 standard deviations above the mean of that distribution ($\mu = 10$). The zscore captures the fact that the score for text #3 was significantly higher than the average score given by subject #1. On the other hand a score of 20 in the second distribution is 0.56 standard deviations below the mean of that distribution ($\mu = 42$) i.e. subject #2 gave text #3 a lower than

average score. If text #3 was a generated text and the other texts were non-jokes in this mini-example, standardizing the raw scores captures the fact that subject #1 considers text #3 funnier than the others whereas subject #2 regards it as less humorous than some of the other texts.

We are interested, not so much in how individual texts measure up against each other, but how generated texts as a genre score in comparison to the other genres in the test (i.e. the human-made New Yorker lexical register jokes, newspaper quotes and proverbs). Therefore, for each volunteer, the 5 or 6 zscores pertaining to generated texts were averaged, as were the zscores pertaining to the three other genres. The volunteer's 20 or 21 answers were in this way reduced to a 4 tuple of averages. 56 volunteers passed the fluency test and so a 56 x 4 matrix of average zscores was produced in which a row represents a volunteer and the 4 columns represent his average zscore for the 4 genres.

An ANOVA was then performed on this matrix to determine whether scores associated with the genres (i.e. the columns of the matrix) are significantly different.

8.4.1 Performing ANOVA and multiple comparison test

A one way repeated measures ANOVA was performed on the genres scores (i.e. the columns of the 56 x 4 matrix described above) and it suggested that there are significant differences between at least some of the genres scores. The significance level was $p < 0.0005$, indicating that there is only a .05% chance that scores this different would arise from randomly selecting from the same (normal) distribution.

ANOVA only suggests whether groups of data are significantly different - it does not indicate which groups are actually different. To determine this, a multiple comparison test was performed. Results of this test suggest that:

1. the New Yorker lexical register jokes are considered significantly funnier than all the other genres.

2. the generated texts are significantly funnier than the non-jokes i.e. the newspaper quotes and proverbs.

3. the scores for the newspaper quotes and the proverbs are not significantly different from each other.

In fact the genres regarded as different from each other were so different that nearly all of them received significance levels of $p < 0.0005$ (The zero values in the "Sig" column of Figure 8.1 are not actually 0: they have been rounded to 0 and they actually signify $p < 0.0005$).

A visual representation of the multiple comparison test, sometimes called an error bar graph, is shown in Figure 8.2. (The ANOVA, multiple comparison test and error bar graph were all performed using the SPSS software). Means are shown as circles in the graph and are significantly different only if their confidence intervals, which appear as vertical lines in the figures, are disjoint. The figure shows that generated texts received significantly lower scores than human made lexical register jokes but significantly higher scores than the non-jokes.

8.4.2 Further analysis of the results

From 203 potential seed texts, 27 texts were generated in a test of the joke generator. These were evaluated and, on average, were found to be significantly funnier than the non-jokes in the test but

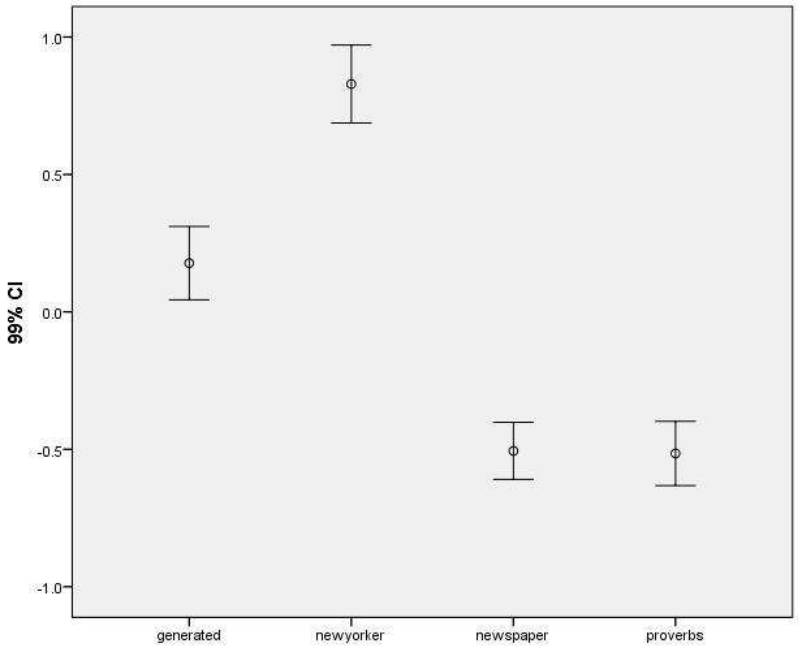
Figure 8.1: Multiple comparison test using Bonferroni correction

Pairwise Comparisons

Measure: MEASURE_1

(I) genre	(J) genre	Mean Difference (I-J)	Std. Error	Sig.[b]	99% Confidence Interval for Difference[b]	
					Lower Bound	Upper Bound
1	2	-.652*	.095	.000	-.965	-.338
	3	.683*	.065	.000	.468	.898
	4	.693*	.076	.000	.441	.944
2	1	.652*	.095	.000	.338	.965
	3	1.335*	.074	.000	1.091	1.579
	4	1.344*	.073	.000	1.101	1.587
3	1	-.683*	.065	.000	-.898	-.468
	2	-1.335*	.074	.000	-1.579	-1.091
	4	.009	.072	1.000	-.229	.248
4	1	-.693*	.076	.000	-.944	-.441
	2	-1.344*	.073	.000	-1.587	-1.101
	3	-.009	.072	1.000	-.248	.229

Figure 8.2: Visualising the multiple comparison test



not as funny as the human-made jokes. Some scores may be misleadingly high, however because the first, second and fourth highest scoring generated jokes (see rows 1, 2 and 4 in Table 8.3) were derived from the same seed text which, in its unaltered form, appears to be somewhat humorous. The seed text is:

> In politics, every day is filled with numerous opportunities for serious error. Enjoy it.

Also, the generated text with the highest score was:

> In politics, every day is filled with numerous opportunities for serious boner. Enjoy it.

The tone of the replacement word 'boner' is informal and does, we believe, humorously contrast with other words in the text such as 'opportunities', 'politics', and 'serious'. However the accidental sexual meaning of the word probably contributes more to the humour of the text than just its informal tone and so the high scores garnered by this text are at least partly the result of chance. The third highest scoring text made the same lexical switch (i.e. changed the word 'error' to 'boner') and its scores are probably also misleadingly high.

Furthermore, the sixth highest scoring text was:

> If you develop commandments, never have more than ten.

The generator replaced the word 'rules' with 'commandments' when generating this text and we believe that the humour of the result is mostly due to an accidental reference to "the ten commandments".

Thus the texts with the top 4 highest scores and the text with the 6th highest score appear to have misleadingly high scores which overestimate the performance of the generator. When these scores are excluded and a second multiple comparison test is performed, the generated texts are still considered significantly funnier than the non-joke texts but not to the same extent as in the earlier test. Figure 8.3 is a visual representation of this second multiple comparison test. As before, means are shown as circles in the graph and are significantly different only if their confidence intervals, which appear as vertical lines in the figures, are disjoint. The mean zscore for the generated texts drops from 0.1775 to 0.0516, bringing it closer to the mean zscores for the non-jokes (-0.4668 for the newspaper texts and -0.4798 for the proverbs) and further away from the mean zscore for the human-made lexical register jokes (0.9099).

8.5 Summary

In development testing of the generator (described in Chapter 7), 2 sets of parameters yielded the best results. Set #1 (shown in Table 8.1) consists of more restrictive parameters which produced only a handful of results in development testing. This was also the case in the final test of the generator: only 2 texts were generated, from 203 potential seed texts, using these parameter values.

The performance of the generator cannot be properly evaluated with so few outputs and so we decided to use the less restrictive parameter settings (set #2 in Table 8.1) in the final test of the generator. From the 203 potential seed texts, 100 texts were generated using these settings and further filtering was applied to these 100 texts, reducing their number to 27. Volunteers were asked to take a fluency test and 56/150 of these people passed the test. These 56 people were then asked

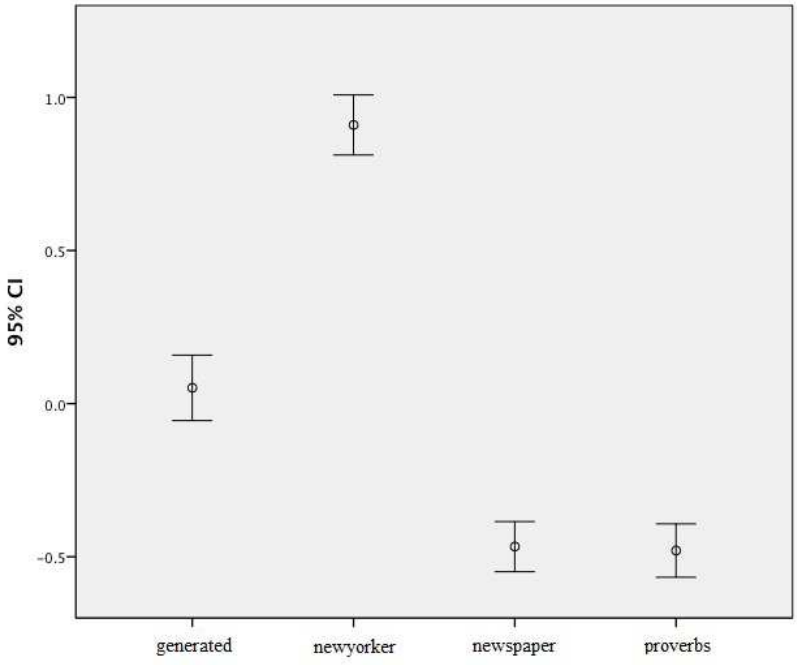
Table 8.2: Generated texts in order of average zscores (from high to low)

no.	generated text	average zscore
1	In politics, every day is filled with numerous opportunities for serious boner. Enjoy it.	1.5445
2	In politics, every day is filled with numerous opportunities for serious misdeed. Enjoy it.	1.235
3	Be precise. A lack of precision is dangerous when the margin of boner is small.	0.9141
4	In politics, every day is filled with numerous opportunities for serious trespass. Enjoy it.	0.8264
5	Can a man take fire in his bosom, and his equipment not be burned?	0.5654
6	If you develop commandments, never have more than ten.	0.4839
7	Can a man take fire in his bosom, and his outfit not be burned?	0.4750
8	Can a man take fire in his bosom, and his ensemble not be burned?	0.4145
9	If you develop propriety, never have more than ten.	0.4049
10	Any change is resisted because bureaucrats have a vested interest in the entropy in which they exist.	0.3258
11	If you develop ordinance, never have more than ten.	0.3112
12	When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian subjection.	0.2298
13	Treat each federal dollar as if it was hard earned; it was by a burgher.	0.1645
14	If you develop precept, never have more than ten.	0.1552

Table 8.3: Generated texts in order of average zscores (continued)

no.	generated text	average zscore
15	Do not speak ill of your predecessors or successors. You did not walk in their sneakers.	0.1128
16	If you develop edict, never have more than ten.	0.0727
17	Arguments of convenience lack righteousness and inevitably trip you up.	.0289
18	Can a man take fire in his bosom, and his tout ensemble not be burned?	-0.0439
19	Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the cosmos.	-0.0798
20	Know that the amount of criticism you receive may correlate somewhat to the amount of promo you receive.	-0.1713
21	Be precise. A lack of precision is dangerous when the margin of misdeed is small.	-0.2587
22	When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian bridle.	-0.2876
23	Any change is resisted because bureaucrats have a vested interest in the tumult in which they exist.	-0.3435
24	Any change is resisted because bureaucrats have a vested interest in the misrule in which they exist.	-0.4529
25	Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the terra.	-0.4588
26	Be precise. A lack of precision is dangerous when the margin of trespass is small.	-0.5563
27	It is necessary for me to establish a winner image. Therefore, I have to beat VIP.	-0.6734

Figure 8.3: Visualising the second multiple comparison test



to score from 0 to 100 how humorous the generated texts and a set of control texts are. An ANOVA and multiple comparison test of these scores revealed that the generated texts are considered not as humorous as human-made lexical register jokes but significantly more humorous than the non-joke texts. This is an encouraging result which suggests that the vector space is somewhat successful in discovering lexical differences in tone and in modelling lexical register jokes.

The following chapter will discuss how the generator might be improved and extended and will explore whether incongruity of tone within a coherent and self-contained text is a sufficient condition for humour.

Chapter 9

Improvements and extensions

This chapter will discuss possible problems with the generator and the vector space and will describe ways the generator might be improved and extended. The least serious issues will be discussed first. We will then examine whether incongruity of tone within a coherent and self-contained text is a sufficient condition for humour.

9.1 Possible problems with the generator

Tables 9.1 and 9.2 list the generated texts according to their average zscores (from highest to lowest). They reveal that only 17/27 of the generated jokes received an average zscore higher than 0 (and two of these were only marginally higher than 0). Thus a high number - at least 10/27 (37%) – of false positives were generated by the system. This section speculates about why the generator output such a high number of non-humorous texts.

9.1.1 Near-synonyms with unclear meanings

Some of the failed jokes output by the generator result from the system's selection and use of near-synonyms with obscure or ambiguous meanings. For example the generator replaced the word 'earth' with the word 'terra' in the following text (generated text #25 in Table 9.2):

> Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the terra.

The meaning of the word 'terra' may be obscure to many readers and perhaps because of this its tone is also unclear. Attardo (1994) suggests that even though the meaning of a word is unknown, it can nonetheless clearly communicate a certain kind of tone. For example the meaning of the word 'electrodynamics' might be unclear but different parts of the word - 'electro' and 'dynamics' - strongly communicate a scientific tone. The same cannot be said of the word 'terra', however. For reasons that are difficult to discern, the word does not, at least according to our intuition, seem to evoke a distinctive tone which contrasts with the tone of other words in the text and its unclear meaning may have something to do with this.

Similarly, the final test of the generator reveals that when the program inserts into a seed text a near-synonym that has multiple and disparate meanings, the tone of such a word can be ambiguous. For example the generator replaced the word 'rules' with the word 'ordinances' in the following text:

> If you develop ordinances, never have more than ten.

Table 9.1: Generated texts in order of average zscores (from high to low)

no.	generated text	average zscore
1	In politics, every day is filled with numerous opportunities for serious boner. Enjoy it.	1.5445
2	In politics, every day is filled with numerous opportunities for serious misdeed. Enjoy it.	1.235
3	Be precise. A lack of precision is dangerous when the margin of boner is small.	0.9141
4	In politics, every day is filled with numerous opportunities for serious trespass. Enjoy it.	0.8264
5	Can a man take fire in his bosom, and his equipment not be burned?	0.5654
6	If you develop commandments, never have more than ten.	0.4839
7	Can a man take fire in his bosom, and his outfit not be burned?	0.4750
8	Can a man take fire in his bosom, and his ensemble not be burned?	0.4145
9	If you develop propriety, never have more than ten.	0.4049
10	Any change is resisted because bureaucrats have a vested interest in the entropy in which they exist.	0.3258
11	If you develop ordinance, never have more than ten.	0.3112
12	When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian subjection.	0.2298
13	Treat each federal dollar as if it was hard earned; it was by a burgher.	0.1645
14	If you develop precept, never have more than ten.	0.1552

Table 9.2: Generated texts in order of average zscores (continued)

no.	generated text	average zscore
15	Do not speak ill of your predecessors or successors. You did not walk in their sneakers.	0.1128
16	If you develop edict, never have more than ten.	0.0727
17	Arguments of convenience lack righteousness and inevitably trip you up.	.0289
18	Can a man take fire in his bosom, and his tout ensemble not be burned?	-0.0439
19	Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the cosmos.	-0.0798
20	Know that the amount of criticism you receive may correlate somewhat to the amount of promo you receive.	-0.1713
21	Be precise. A lack of precision is dangerous when the margin of misdeed is small.	-0.2587
22	When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian bridle.	-0.2876
23	Any change is resisted because bureaucrats have a vested interest in the tumult in which they exist.	-0.3435
24	Any change is resisted because bureaucrats have a vested interest in the misrule in which they exist.	-0.4529
25	Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the terra.	-0.4588
26	Be precise. A lack of precision is dangerous when the margin of trespass is small.	-0.5563
27	It is necessary for me to establish a winner image. Therefore, I have to beat VIP.	-0.6734

The word ‘ordinance’ is a somewhat archaic and formal synonym for ‘rule’ or ‘law’ but it can also mean ‘bomb’ and these significantly different meanings evoke different kinds of tone. It is unclear which of these two meanings is intended in the text output by the generator and so the tone communicated by the word is similarly unclear. This confusion of tones prevents the word from generating, according to our intuition, a clear incongruity in the text.

9.1.2 Confusing changes to common expressions

In two of the texts output by the generator (texts #21 and #26 in Table 8.3) the program changed the phrase “margin of error” to “margin of trespass” and “margin of misdeed”. The words ‘trespass’ and ‘misdeed’, however, are poor choices as near-synonyms for the word ‘error’ as it is used in the expression “margin of error”. The resulting phrases sound odd and are somewhat difficult to understand and the confusion they create may interfere with the reader’s appreciation of incongruities in the text. If the joke generator could anticipate whether a given lexical change will interfere too much with the meaning of a text, better results might be achieved. Equipping a system with this ability is a difficult and unsolved problem in semantics however.

It is important to note that altering a common expression will not always render it confusing and result in a failed joke. Section 9.5.4 discusses how the generator might be extended to generate a type of register humour in which common expressions are altered to create humorous incongruities of tone.

9.2 Possible problem with the vector space

Corpus sparsity will result in inaccurate frequency profiles and flawed estimates of tone and this is probably the biggest problem affecting the vector space classifiers and the generator. Profile A and C words (see Chapter 4 for details) are an obvious manifestation of this problem and the generator mitigated the effect of such words by implementing a corpus threshold: texts containing profile A or C words were not used as seed texts and near-synonyms with such profiles were not inserted into seed texts. In the final test of the generator, none of the texts output by the system contain such words. All the words in the generated texts appear in no fewer than 4 corpora. (The only exception is the word ‘promo’ which occurs in three corpora - the two corpora of rap lyrics and the corpus of movie reviews).

Corpus sparsity does not simply produce problematic profile A or C words, however. A word might appear in numerous corpora but its frequencies in some or all of the corpora in which it appears could be misleadingly low or high because of inadequate corpus coverage. This too can result in faulty frequency profiles and impair the system’s estimates of tone. Adding more corpus data or using Google n-gram information (Michel et al., 2011) could improve the accuracy of frequency profiles and enhance the classifiers’ and generator’s performance.

Another way of addressing the corpus sparsity problem would be to compute the frequencies of lexemes rather than words. Hash tables of words and their normalized frequencies in the various corpora were computed and the classifiers and generator make use of these tables to construct the frequency profiles of words. A better idea would have been to create hash tables of lexemes (i.e. root forms of words rather than surface forms) and their frequencies as this would improve corpus coverage.

9.3 Possible problem with the model - incongruity of tone not sufficient

Another explanation for poor generator output may be that incongruity of tone is insufficient for humour. The experiments described in Nerhardt (1970) suggest that incongruity alone is sufficient for humour but this does not appear to be true in the case of lexical register jokes. When we examine the test set of lexical register jokes (shown in Table 9.3), for instance, it appears that incongruity generates humour only when it is applied in a certain way. This hypothesis aligns with Oring's analysis of humour which states that incongruity is not humorous in itself - that "there are numerous instances of incongruities that generate no humour" – and that incongruity has to be used in a certain way to create humour (Oring, 2003). For example Oring points out that "definitions and metaphors are rooted in ... incongruities yet neither definitions nor metaphors are in themselves funny". Metaphors "proclaim that one thing is like another; indeed that something is something other than itself". Oring asserts that this is "a logically absurd proposition" and definitions are similar to metaphors in this regard because they also describe concepts in terms of other concepts. For instance Aristotle defined man as a "rational animal" and, like all definitions, the statement is incongruous:

> Man and animal are incongruous categories because humans are rational while animals are not. Yet human and animal are appropriately related because a human being is an animal in all remaining characteristics (Oring, 2003).

Like jokes, metaphors and definitions house incongruous ideas yet unlike jokes, these constructions are not humorous and Oring argues that the reason for this is that "in jokes the engagement of the incongruity and the search for its appropriateness is spurious rather than genuine".

Oring's argument that incongruity is not humorous in itself but has to be used in a certain way to create humour is convincing and a close examination of lexical register jokes seems to support this assertion. It is unclear, however, whether Oring is suggesting that incongruity is humorous only when it reveals spurious thinking or whether he would argue that this is just one way incongruity can be used for humorous effect. We would argue the latter because when the test set of lexical register jokes is examined, for example, we find that the incongruity creates a humorous effect, not because it reveals faulty reasoning, but because it challenges a stereotype or undermines someone or something that possesses a kind of social power.

In jokes #1-3, businessmen are described as naughty children, Caesar and his army are depicted as surfer boys fighting a silly war and the social position of a cultured 'foodie' is tainted when she is associated with street talk – the language of people who possess little power and status. In jokes #4-6, the seriousness of academia is satirized, the power of television executives is greatly exaggerated (and therefore undermined), and the archaism and irrationality of prayer is juxtaposed with a more prosaic kind of decision-making. In the next three jokes, the superficiality of fame in the modern age is contrasted with the idea of honour in more chivalrous times, the vanity and self-consciousness of a father are suggested and the free-spiritedness and rebelliousness of a revolutionary group is undermined when it is shown possessing some of the same characteristics as the systems of power it opposes. Finally, in the last two jokes of the test set, the pretensions of the art world are gently mocked and the power and prestige of the American presidency are

Table 9.3: The test set of lexical register jokes

1	This cell block is for naughty businessmen like yourself who were caught price fixing and such.
2	Hail Caesar! The enemy has been totalled.
3	The woman who ordered the poulet chasseur would like to come in and rap about it.
4	If you do not mind my saying so Doctor, 'Exegetical Dissertations on Theosophical Redirections in the Twentieth Century' will not have quite the impact of 'Keep the Faith Baby!'.
5	Woe! The gods have now decreed that there will be a short pause for a commercial.
6	And finally Lord may thy wisdom guide and inspire the deliberations of those thy servants our planning and zoning commission.
7	I deem thee newsworthy.
8	You are getting to be a big boy sonny and when we are alone it is still nice to hear you call me Popsy. But in front of people try to remember to call me Sire.
9	Miss would you type this searing blast at the honkie power structure in triplicate for me please?
10	There is a skillful juxtaposition of contrasting elements but doggone it I miss his oblique offhand symbolism.
11	I warned you that giving him the presidency would be a boo boo.

undermined.

Thus incongruity of tone in lexical register jokes may be a means to an end and perhaps should not be regarded as an end in itself. This aligns with Oring's theory that how incongruity is used is important and determines whether it generates humour or not. However incongruity in lexical register jokes appears to be humorous, not because it is used to suggest spurious thinking but because inferences arising from it attack a stereotype or undermine something powerful.

Lexical register jokes, and register humour in general, are similar to the children's books in which characters' clothing can be made to clash in humorous ways. The books are split in a way that allows the reader to mismatch outfits so that a figure can be shown wearing a chimeric ensemble such as a fireman's hat, a nurse's uniform and ballerina slippers, for example. Conventions of fashion dictate that the various parts of an outfit should be consistent in style and colour and when these conventions are flaunted, the result can be humorous. Similar conventions occur in speech and writing. The tone of words in a text normally conform to a consistent and overarching style but when this convention is defied, the results, like the children's books, can be amusing.

Interestingly, outfits in the children's books are always shown on someone because the outfits themselves may not be inherently humorous. The clashing outfits are perhaps only funny when worn by a person because they make that person look ridiculous and the same may be true about conflicts of tone in register humour. Incongruity of tone may not be humorous in itself and may only be amusing when it undermines a person's dignity.

Veale (2004) addresses this issue and suggests that incongruity may not be an essential ingredient of humour, as many theories of humour suggest (Attardo, 1994; Ritchie, 2004) but is merely

'epiphenomenal' i.e. a secondary phenomenon that often occurs in jokes but is not primarily responsible for their humour. Veale argues that to explain why a text is funny, "a humour theory must not look to incongruities but provide a social explanation for why we enjoy insulting others".

One of the jokes Veale examines in the paper is the following, which appears in (Freud, 1966):

> Serenissimus was making a tour through his provinces and noticed a man in the crowd who bore a striking resemblance to his own exalted person. He beckoned to him and asked: "Was your mother at one time in service at the palace?" - "No your highness", was the reply, "but my father was".

Veale argues that the humour of the passage seems to result from a collaboration between the writer of the text and its readers to "lay low a figure of privilege and authority". Veale points out that the punch line does not force a reader to the humorous conclusion that Serenissimus, rather than the peasant, is the bastard child because there are "at least two alternate construals of the punchline that are not humorous: (1) The peasant's father did work in the palace, but no affair with the queen took place. (2) The physical resemblance with Sernissimus is merely a quite reasonable coincidence". Instead of choosing one of these non-humorous interpretations, the reader selects an interpretation which is "inherently humorous" because it undermines the dignity of an authority figure. Veale argues that the writer and reader, in "a manner consonant with the Freudian view of humour", seem to "collaborate together to achieve a socially licensed spot of ego-popping" (Veale, 2004). Veale argues that a reader's "social conditioning" is such that he finds it "gratifying to see narratives where pomposity is deflated, excessive authority is thwarted, modesty is reward and arrogance is punished". Readers are therefore 'instinctively' primed to search for interpretations of texts which undermine authority.

The collaboration between writer and reader described by Veale may be the core mechanism at work in lexical register jokes. Incongruity of tone may not be inherently humorous but might instead act as a kind of opportunity for the reader to search for a humorous interpretation of the text. When a reader encounters a clash of tone in a text, he searches for an explanation for it because texts normally maintain a uniform tone. If an explanation can be found in which pomposity is deflated or, conversely, a normally marginalized entity is granted power, the text elicits a humorous response. On the other hand if no such interpretation can be found, the clash of tone may fail to ignite into humour.

9.4 Further testing of the model

Further tests of our model, such as those listed below, might help determine whether incongruity of tone is sufficient for humour or whether other features need to be present in a text.

9.4.1 Provide attribution

Seed texts on which the generated texts were based consisted of biblical proverbs and quotes from famous politicians such as Donald Rumsfeld, Barrack Obama, and Richard Nixon. However, this attribution information was not provided to judges who evaluated and scored the texts output by the generator. It would be interesting to compare the scores from that evaluation with those in an evaluation in which attribution of the original quotes was provided. If the latter scores are higher,

this might support the argument (made in Section 9.3) that lexical register jokes need to evoke (and undermine) a personality in the text. The test would not be definitive in this regard, however. Providing attribution would introduce or add more of a personality into a text but it would also provide more context. Providing more context might make the quoted texts more self-contained and, as suggested in Chapter 7, self-containment also seems to be an important feature of lexical register jokes. If providing attribution were to improve scores, it would be unclear whether this was because texts were more self-contained or because more of an identifiable personality had been injected into the text. The fact that scores are higher because one or both of these features were present would nonetheless be important and perhaps further testing could determine the ultimate cause.

9.4.2 Make identical change to different texts

Another interesting test would be to make identical near-synonym replacements in multiple texts. A number of self-contained regular texts containing the word 'wicked', for example, might be found and the word changed to a near-synonym with significantly different tone such as 'naughty'. Texts would be scored by human judges to determine whether the different contexts in which the word change was made had a significant effect on the scores.

Words in seed texts could perhaps be annotated with various tags from the General Inquirer Lexicon (GIL) (Stone, 1997). GIL tags such as HUMAN and POWER for example indicate if a word is somehow associated with human beings or has connotations of power - two semantic features that might, as argued in Section 9.3, be important in lexical register jokes. Tags from the MRC database which estimate the concreteness or imageability of words might also be used (Wilson, 1987). For example the generated text "Do not speak ill of your predecessors or successors. You did not walk in their sneakers." was produced by replacing the word 'shoes' with 'sneakers' and this text may be more humorous than some of the other generated texts because of how tangible or concrete the target word 'shoes' is. (What a person wears is often tied closely to their personality so perhaps the human or personal dimension of the word is also important here for the reasons given in Section 9.3). Making identical near-synonym replacements in texts that have different GIL and MRC annotations might reveal whether other features, besides incongruity of tone, are important for lexical joking.

9.5 Extending the classifier and generator

9.5.1 Syntactic tests

The quality of the texts output by the generator could be improved by performing some simple syntactic tests. For example texts such as "If you develop commandment, never have more than ten" could be improved by enforcing number agreement.

9.5.2 Use Wordnet and a backing-off strategy

Wordnet (Miller, 1995) could be used as an alternative to the resource currently used by the generator (www.thesaurus.com) for retrieving synonyms. The thesaurus.com website was used because it appeared to contain more synonyms which vary in terms of their tone. Wordnet, however, could be used to expand the search for not just near-synonyms of words. More general (hypernyms) or more specific (hyponyms) words with different tone could be used by the generator and evaluated.

9.5.3 Use LSA

Latent Semantic Analysis (LSA) "represents the meaning of a word as a kind of average of the meaning of all the passages in which it appears, and the meaning of a passage as a kind of average of the meaning of all the words it contains" (Landauer et al., 1998). LSA can be used to estimate the similarity of sentences or larger passages and perhaps the generator could make use of this method to automate the search for promising seed texts. Seed texts that are semantically similar to the funniest human-made lexical register jokes (or to seed texts that were previously used by the generator to construct the funniest generated lexical register jokes) could be automatically found and used to construct new lexical register jokes. In this way the system could remain agnostic about what semantic features, besides incongruity of tone, are necessary for lexical joking. (A qualitative analysis of the seed texts deemed similar to the best lexical register jokes could be performed, however, to determine what these other semantic features might be).

9.5.4 Altering common expressions

Common expressions might be especially fertile ground in which to create clashes of tone. For example in the following human-made examples of register humour, all of which are captions from New Yorker cartoons, a word in an expression has been replaced to create a humorous conflict of tone:

> "Farewell cruel environment".
>
> "Sticks and stones may break my bones but rhetoric will never hurt me.
>
> "God bless our module".
>
> "Was it something I averred?".

In these texts the word 'environment' replaces the word 'world', 'rhetoric' replaces 'words', 'module' replaces 'home', and 'averred' replaces the word 'said' but the altered expressions are still comprehensible and the resulting texts are humorous. Interestingly, the tone of the replacement word conflicts with the tone of the word it has replaced. The absent paradigmatic word is evoked because the expression is so well known and it participates in creating incongruity in the text in spite of its absence.

To create or recognize these kinds of jokes, a system would:

- have to be able to recognize that a piece of text is an expression (perhaps by computing co-occurrence statistics of words on the web).
- need to be able to estimate the tone of expressions and idioms. (Considerably more corpus data would be required for this. The current system suffers from problems of corpus sparsity when handling just unigrams and providing enough coverage for ngrams (where n > 1) would be even more of a challenge).
- need to plot the word that was replaced (word A), the word that replaces it (word B), and the expression as a whole to determine if word B clashes with both word A and the original (unaltered) expression.

- need to anticipate whether the change to an expression is too semantically disruptive. (For instance, in the final test, the generator output two lexical register jokes (texts #21 and #26 in Table 8.3) that appear to have failed because changes to an expression obscured its meaning).

As discussed in Section 9.1.2, implementing the last item in the list above would be particularly challenging and may not be possible right now.

9.5.5 Create captions for cartoons

It would be interesting to build a system in which one kind of tone is represented in a cartoon and an opposing tone is generated in a caption. Numerous examples of this kind of register humour can be found in the New Yorker. One cartoon, for example, shows two conservatively dressed middle-aged women asking a museum guard whether he is "packing a heater" (i.e. carrying a gun). In another cartoon, two homeless are drinking on a park bench and one says to the other: "I thought that was rather a snivelling little muscatel didn't you?". Another cartoon shows a doctor saying to his patient, a well-dressed businessman: "And when did you first notice that your tummy was not behaving?".

Using a cartoon to communicate one kind of tone, which is opposed in the caption, can be effective because a cartoon can:

- provide structure for a joke.
- clearly and economically communicate different kinds of tone without burdening the reader with extraneous details (i.e. details which could distract the audience from potentially humorous oppositions).
- provide concreteness to a joke. As mentioned in Section 9.4.2, creating incongruities of tone with concrete objects might create better lexical register jokes.
- provide a context or vignette or scene which is self-contained. As argued in Chapter 7, self-containment seems to be an important feature of lexical register jokes and this may be true of other kinds of register humour.

Currently it would be too difficult to equip a system to recognize characters and objects in a cartoon or image and determine what kind of tone they evoke. However this kind of information might be gathered from people in an electronic form, for example, and used by a generator to create a caption for the image.

Chapter 10

Summary

Many theories of humour claim that incongruity is an essential ingredient of humour (for example Koestler (1964); Keith-Spiegel (1972); Suls (1983)) but the concept of humorous incongruity is poorly understood. In fact linguistic research has failed even to construct a precise definition of this concept (Ritchie, 2004).

Computational linguistics brings a measure of detail and precision to the study of humour because careful analyses of the mechanisms of humour have to be made to enable algorithmic generation or detection of humorous material. In the past, joke generators and classifiers have been built which attempt to model certain subclasses of humour and developing these systems has forced investigations of humour to be more rigorous and detailed.

Even so, such systems tend to produce inconsistent results because, in spite of their increased attention to detail, none implement a model of incongruity. For example Ritchie argues that the pun generators (such as those described in Lessard and Levison (1992, 1993, 2005); Binsted (1996); Manurung et al. (2008)) do not implement a model of humorous incongruity and that "if humour results from any 'incongruity' in the output texts, this is purely fortuitous" (Ritchie, 2004). Similarly classifiers such as those described in Mihalcea and Strapparava (2006) , Mihalcea and Pulman (2007) attempt to distinguish between humorous and non-humorous texts by looking for secondary features of humorous texts rather than for incongruity, and perhaps for this reason, results of these systems are also mixed.

As in previous research, this thesis develops classifiers and a joke generator which attempt to automatically recognize and generate a type of humour. However the systems described in this thesis differ from previous programs because they implement a model of a certain type of humorous incongruity.

10.1 Lexical register jokes

The type of humorous incongruity modelled in this thesis occurs in a subclass of register humour we call lexical register jokes. These are humorous texts in which a conflict of tone occurs between individual words (rather than phrases or the syntax of a sentence). The kind of tone at play in lexical register jokes is hard to define, however. Intuitively it seems to be composed of various dimensions such as formality, archaism, literariness, and how flowery a word is but whether it is precisely this combination of dimensions (some of which do not seem to be independent of each other) or some other combination, is difficult to discern.

For this reason we decided that a way of specifying the property we are interested in would be

to make use of example corpora that seem to vary on it. Corpora that vary in the kind of formality and informality of tone we think is at play in lexical register jokes were assembled and these corpora form the basis of a semantic space. Frequencies of a word across the different corpora are computed and together they act as an estimate of the word's tone. These frequencies can be regarded as coordinates in a multi-dimensional space and this space forms the backbone of the joke generator and most of the classifiers developed in this thesis.

Pun generators such as those mentioned above bring homophones together and attempt to build plausible contexts for them. But they do so without first determining whether the ideas evoked by the homophones are humorously incongruous rather than simply different. And so the resulting clash of ideas is sometimes odd rather than humorous. Our goal was to create a semantic space that reflects the kind of tone that is at play in lexical register jokes so that words that are far apart in the space are not simply different but exhibit the kinds of incongruities of tone we see in lexical register jokes.

10.2 Performance of the systems

Classifier #4 (introduced in Chapter 5) plots the words of a text into the vector space and measures the distance between each pair of words. If any of these values fall above a certain threshold, the text is classified as a lexical register joke. This classifier yielded the highest precision scores in final tests on unseen data. It achieved an average accuracy of 88.7% when distinguishing between a test set of lexical register jokes and 4 different kinds of regular text (two sets of newspaper quotes, a set of randomly selected proverbs from the bible and a set of randomly selected captions from New Yorker cartoons which are not lexical register jokes). These results were encouraging.

However in a test on unbalanced data (Chapter 6) all the vector space classifiers, including classifier #4, yielded a high number of false positives (i.e. regular texts mistakenly classified as lexical register jokes). The most likely reason for this is that the corpora used to create estimates of tone are a finite resource and do not provide perfect coverage.

Universally poor precision scores in this classification test on unbalanced data had important implications for building a joke generator that uses the vector space to gauge differences in tone. Low scores in this test suggest that such a generator is likely to produce a significant number of texts which are not in fact lexical register jokes. Filtering mechanisms which attempt to minimize this problem were therefore implemented into the joke generator (see chapter 7 for details).

From 203 potential seed texts, 27 texts were created by the generator and presented as lexical register jokes. Volunteers were asked to score, from 0 to 100, how humorous the generated texts and a set of control texts were. The generated texts were considered not as humorous as human-made lexical register jokes but significantly more humorous than the non-joke texts. This was an encouraging result which suggests that the vector space is somewhat successful in discovering lexical differences in tone and in modelling lexical register jokes.

The joke generator was not entirely successful however, and corpus sparsity was still the most likely reason for the mixed results. Filtering mechanisms which were implemented to address this problem could eliminate only the most obvious errors arising from corpus sparsity and an imperfect vector space continued to impair estimates of tone. Adding more corpus data to the system is an obvious and increasingly viable solution to this problem as the number of machine

readable texts being produced continues to grow.

Another possible factor affecting the generator's performance is that incongruity of tone is not a sufficient condition for humour (see Chapter 9 for details). It may be the case that incongruity of tone generates humour only when it manages to undermine someone's dignity or to reverse a stereotype.

10.3 Conclusion

The research described in this thesis differs from previous work in computational humour in a number of ways. It is the first to model a type of register humour and to use word frequencies as a measure of register. Register humour is perhaps more subtle than the kind of humour previously studied in computational research. It appears to be less formulaic and more dependent on context than punning, for example, and consequently may be a more challenging type of humour to model.

Perhaps most importantly, however, this thesis differs from previous research because it attempts to objectively quantify and measure a type of humorous incongruity. Previous joke generators "abstract[] structural patterns from existing examples, formulate[] computationally manageable rules that describe these patterns and implement[] an algorithm to handle these rules" (Ritchie, 2009a) but none of these systems implement a model of a type of humorous incongruity.

The research in this thesis builds and implements a model of a type of humorous incongruity, achieves some success in automatically recognizing and generating a subclass of humour, and suggests perhaps that other types of humorous incongruity can also be quantified.

Appendix A

The corpora used in corpus sets A - E

Chapters 3 and 4 refer to this appendix. The sources of the corpora are:

- OTA: The Oxford Text Archive. http://ota.ahds.ac.uk/
- PG: Project Gutenberg. http://www.gutenberg.org/wiki/MainPage
- OMCS: OpenMind Common Sense statements. http://commons.media.mit.edu, http://openmind.media.mit.edu/
- NLTK: The Natural Language Toolkit. http://www.nltk.org/
- CUVP: The CUVPlus dictionary provides BNC frequency counts directly.
- LDC: Linguistic Data Consortium

A.1 Corpus set A

Table A.1: Corpus set A

no.	corpus	no. of words	source
1	Virgil's The Aeneid (17th century translation)	108,677	OTA
2	all of Jane Austen's novels (early 19th century)	745,926	OTA
3	King James bible (17th century translation)	852,313	PG
4	All of Shakespeare's tragedies and comedies (1623 first folio edition)	996,280	OTA
5	Grimm's fairy tales (19th century)	281,451	PG
6	All the poems of Samuel Taylor Coleridge (early 19th century)	101,034	OTA
7	Two novels by Henry Fielding (18th century)	148,337	OTA
8	Collection of common sense statements	2,215,652	OMCS
9	A corpus of Reuter's new articles	1,614,077	NLTK package
10	Science articles	366,662	NLTK package
11	Movie reviews	1,298,728	NLTK package
12	The written section of the British National Corpus (World Edition).	80 million	BNC word frequencies from CUVPlus dictionary, OTA

A.2 Corpus set B

Corpus set B was built by adding the following four corpora to corpus set A:

Table A.2: Corpus set B adds the following corpora to corpus set A

no.	corpus	no. of words	source
1	Thomas Bulfinch (19th century)	222,462	OTA
2	Alexander Pope's translation of Homer's The Odyssey (18th century) and Samuel Butler's translation of The Iliad (19th century)	265,068	PG
3	Poetry by John Keats (early 19th century)	113,665	OTA
4	Poetry by John Milton, including Paradise Lost (17th century)	143,522	PG

A.3 Corpus set C

Corpus set C was created by adding the following four corpora to corpus B:

Table A.3: Corpus set C adds the following corpora to corpus set B

no.	corpus	no. of words	source
1	Sir Walter Scott's Ivanhoe (19th century)	188,740	OTA
2	Science essays by British students (present day)	483,451	OTA
3	A corpus of informal blogs (present day)	386,388	created using SketchEngine http://www.sketchengine.co.uk/
4	A corpus of documents about physics (present day)	369,554	created using SketchEngine http://www.sketchengine.co.uk/

A.4 Corpus set D

We combined 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, Henry Fielding and John Keats were removed because they seemed to be relatively less extreme exemplars of formality than the others. These changes resulted in corpus set D:

Table A.4: Corpus set D

no.	corpus	no. of words	source
1	Virgil and Homer's works	378,568	OTA + PG
2	Shakespeare and Coleridge's works	1,112,925	OTA
3	Bulfinch and Scott's works	412,660	OTA
4	King James bible (17th century translation)	852,313	PG
5	Grimm's fairy tales (19th century)	281,451	PG
6	Common sense statements	2,215,652	OMCS
7	A corpus of Reuter's new articles	1,614,077	NLTK package
8	Science articles	366,662	NLTK package
9	Movie reviews	1,298,728	NLTK package
10	The written section of the British National Corpus (World Edition).	80 million	BNC word frequencies from CUVPlus dictionary, OTA
11	Poetry by John Milton, including Paradise Lost (17th century)	143,522	PG
12	Science essays by British students (present day)	483,451	OTA
13	Arts essays by British students (present day)	324,042	OTA
14	A corpus of informal blogs (present day)	386,388	created using SketchEngine http://www.sketchengine.co.uk/
15	A corpus of documents about physics (present day)	369,554	created using SketchEngine http://www.sketchengine.co.uk/

A.5 Corpus set E

Corpora of rap music lyrics, transcripts from the television show South Park, New York Times books reviews and science articles were added to corpus set D to create corpus set E. Also the corpora created using the SketchEngine tool (i.e. corpus of informal blogs and a corpus about physics) were removed because they were assembled from the web by a program whose algorithm was unknown to us and because the exact sources of the contents of these corpora were unknown. Corpora set E thus contains 20 million words - roughly twice as much data as corpora set D:

Table A.5: Corpus set E

no.	corpus	no. of words	source
1	Virgil and Homer's works	378,568	OTA + PG
2	Shakespeare and Coleridge's works	1,112,925	OTA
3	Bulfinch and Scott's works	412,660	OTA
4	King James bible (17th century translation)	852,313	PG
5	Grimm's fairy tales (19th century)	281,451	PG
6	Collection of common sense statements	2,215,652	OMCS
7	A corpus of Reuter's new articles	1,614,077	NLTK package
8	Science articles	366,662	NLTK package
9	Movie reviews	1,298,728	NLTK package
10	The written section of the British National Corpus (World Edition).	80 million	BNC word frequencies from CUVPlus dictionary, OTA
11	Poetry by John Milton, including Paradise Lost (17th century)	143,522	PG
12	Science essays by British students (present day)	483,451	OTA
13	Arts essays by British students (present day)	324,042	OTA
14	New Scientist articles	854,892	OTA
15	Overheard statements	160,431	NLTK package
16	South Park episodes (seasons 1-6)	419,622	http://www.spscriptorium.com/
17	South Park episodes (seasons 7-11))	398,409	http://www.spscriptorium.com/
18	Book reviews from New York Times (1987)	1,672,248	New York Times Annotated Corpus (LDC)
19	Book reviews from New York Times (1988)	1,675,977	New York Times Annotated Corpus (LDC)
20	Science articles from New York Times (1987-1988)	1,298,059	New York Times Annotated Corpus (LDC)
21	Science articles from New York Times (2005-2006)	844,345	New York Times Annotated Corpus (LDC)
22	Speeches and selected writings by Winston Churchill	473,816	PG
23	All the novels by the Bronte sisters (19th century)	1,063,370	PG
24	Rap lyrics (part 1)	1,106,950	from http://www.ohhla.com
25	Rap lyrics (part 2)	1,104,362	from http://www.ohhla.com

Appendix B

The regular texts of the development and test sets

Chapter 3 refers to this appendix.

B.1 Development set #1 of newspaper texts

(from the June 5 2009 issue of the Globe and Mail, a Canadian national newspaper)

1. The tide of job losses washing across north america is showing signs of ebbing, feeding hope that
2. Yet investors and economists are looking past the grim tallies and focusing on subtle details that suggest
3. There is definitely dancing at the prom an annual rite for muslim teens but no boys, no
4. It is a shining example of the intersection of cultures that the president stressed in his historic
5. He insisted last night that he would not waver or walk away from power at the end
6. Members of both are forever claiming public interest and good as the end goal while rarely declaring
7. It was also pointed out that some of the ministers with the most controversial expenses claims, details
8. Almost always the only completely opaque cost is the cost of legal aid. Lawyers oppose such disclosure
9. The chance to apply for early parole after serving so many years behind bars would be to deny
10. A panel of international experts is recommending that the way diabetes is diagnosed should be dramatically simplified
11. Bumping into a member of the British royal family was an unexpected bonus for the visitors who
12. Even if the resolution does not pass, it has succeeded she said in shining a spotlight on

13. The resolution reflects growing dismay among municipalities over being shut out of lucrative infrastructure jobs as a
14. One of the dominant phenomena in the art world of the past three decades or so has
15. The exhibition is a historic event formally linking at last the planet two leading centres for the
16. At thursday night performance the boy beside me could not have been more than actually shrieked in
17. Both runs were completely sold out and he was so mobbed at the stage door that he
18. The bid has already been won. what is to stop from squeezing the architect for cuts to
19. Advertising executives packed into the hall for television presentation a rite of spring passage in the world
20. He contorts a bit, raising his right shoulder wringing one hand with the other and fingering his

B.2 Development set #2 of newspaper quotes

(from the November 29 2010 edition of the Canadian Broadcasting Corporation website)

1. We need to work with the family in making sure they find the objects, and we need
2. The secretary general looks forward to a solution to the political crisis in the country since any
3. This new drug is much more effective than the current therapy, and safer, and on top of
4. It is important not only just to have the adverse drug reaction, but also more information about
5. As a mom, I am proud that our new, tough regulations will make a world leader in
6. What I can say is that if the chief has relevant information that will assist us in
7. In order for us to close the file, we have to complete the medical exams to make
8. We have no way of knowing what has been removed. It is very likely that what has
9. This collaboration between new government and the Foundation is going to contribute to the global effort to
10. I have seen every change in this business, and I think this one is the most dramatic
11. An attack is an attack, whether it is large or small, and we are trying to defeat
12. It is the longest ransom note in history do what we tell you and you may, in
13. As the program begins to work, we would expect that would be able to go back into

14. The attack on patent holders and the adverse implications from the standard is proposing is unprecedented and

15. The hard truth is that getting this deficit under control requires some sacrifice, and that sacrifice must

16. I thought, if I had to let my baby die to get into heaven, maybe heaven is

17. I think things will probably just move forward. I do not know if the international community will

18. If we could just get everybody to eat one more pound of apples per year that would

19. We totally denounce the use of violence to achieve anything. It is not the way, it is

20. We take his concerns seriously when he raises them so I would like an opportunity to discuss

B.3 Test set #1 of newspaper quotes

(validated quotes taken from the November 29 2010 issues of the Canadian Broadcasting News (CBC) and the Globe and Mail news websites)

1. I got the okay to just take the wood and, to me, it was just a pile of wood.

2. It really is the poster child, unfortunately, for the kinds of changes that are going to have to happen.

3. Leadership in sports, like in life, business, or politics, always comes down to the same ingredients.

4. It has an institutional dimension that cannot be ignored given the circumstances.

5. The personal computer was designed for people to do stuff with it that the inventors of that device had never even dreamed of.

6. We loved him dearly and we will miss him and he was a good friend of mine.

7. This is about bringing respect for taxpayers back to City Hall. This team I have demonstrated here will do that.

8. We do things in order to get infrastructure legacy projects. That is what we do it for.

9. She enjoyed outings, one of which was to the circus, where she smiled when the horses went by.

10. The evidence that they are relying on is false. It has been edited. A significant portion of it has been removed.

11. The mayor himself has talked about waste, and we just think it is way out of line, the amount.

B.4 Test set #2 of newspaper quotes

(unvalidated quotes taken from the February 8, 2012 paper version of the Canadian newspaper the Globe and Mail)

1. We heard that they were sent to the hospital. That was when we heard that he did not make it.
2. The thing that is nice about pregnancy is that in the end you have a baby.
3. It is going to be important for us to think about what our reliance on non renewable resource revenues will be.
4. Of all the candidates he is the one who is the least at ease in French, which is a fundamental problem.
5. He would rip off the wings of all the angels in heaven and sell them to the devil for his own gain if he could.
6. If I did not have people to pick my crops, I would not need my truck driver.
7. For all these reasons workers are easy targets in foreign lands.
8. I believe if I continue as the president, the people of the country would suffer more. I wish they would have a consolidated democracy.
9. It is really the material I use that dictates a certain kind of mood and the connections I can make between the footage.
10. You can not apologize for taking a position in a debate because otherwise you would never take a position in a debate.
11. I did not think there were too many members who did not connect the twin events, me being admitted to membership with the spontaneous fire.

Appendix C

How we clustered the lexical jokes by hand

Chapter 5 refers to this appendix.

C.1 The development set of simple lexical jokes.

A simple lexical joke is a text in which the tone of a single word (tone A) conflicts with the tone of one or more other words in the text (tone B) and both tone A and tone B are presented lexically. In the list of jokes below, the word with tone A is shown in bold and word(s) with tone B are underlined.

1. Operator, I would like to make a **personage** to person call please
2. Mom! Bart is on a strict diet of complex carbohydrates. Steak will make him **logy**.
3. Thirty two years as a chef and he can still go **yum**.
4. Cancel my appointments for this afternoon miss. I am **rapping** with my son.
5. Gentlemen - nothing stands in the way of a final accord except that management wants profit maximization and the union wants more **moola**.
6. Well all our data confirm your own original diagnosis. You are just plain **tuckered** out.
7. Sometimes I think you are a serious research and development man and sometimes I think you are just **messing** around.
8. You cannot expect to wield supreme executive power just because some watery **tart** threw a sword at you.
9. Oh, how could I fall for fake Superbowl tickets? Gee, the fellas are gonna be **crestfallen**.
10. **Gee**! Determining which issues have growth potential while simultaneously working to provide your clients with a reasonable annual yield is most certainly creative.
11. It is best not to use big words. Why choose a big word when a **diminutive** one will do?
12. Damn it, agree to whatever she demands no matter what it takes I want my **mommy**.
13. Listen, serving the customer is **merriment** enough for me.
14. Last chance to see the beautiful reflections in Mirror Lake before **eutrophication** sets in.

15. The market gave a good account of itself today, **Daddy**, after some midmorning profit taking.

16. I understand you perfectly. When you say you want to extend your parameters, it means you want **floozies**.

17. Thou shalt not horn in on thy husband's **racket**.

18. Why art thou giving me a hard time? **Eh**? Speak up!

19. Forbearance is the watchword. That triumvirate of **Twinkies** merely overwhelmed my resolve.

20. His grace the lord archbishop has granted me an audience tomorrow. Work up a few **zingers** will you?

C.2 Development set of more complicated lexical jokes

1. Would you care to know dear that your hazy **morn** of an **enchanted** day in May is composed of six tenths parts per million sulfur dioxide, two parts per million carbon monoxide, four parts per million hydrocarbons, three parts ...

2. And then he strode into the board meeting **brandishing** this flaming sword and said **Woe unto** him who moves his corporate headquarters out to the suburbs.

3. And finally **Lord** may **thy** wisdom guide and inspire the deliberations of those **thy** servants our planning and zoning commission.

4. Oh, **Lord**, bless this **thy** hand grenade that with it **thou mayest** blow **thy** enemies to tiny bits, in **thy** mercy.

C.3 Test set of simple lexical jokes

1. This cell block is for **naughty** businessmen like yourself who were caught price fixing and such.

2. Hail Caesar! The enemy has been **totalled**.

3. The woman who ordered the poulet chasseur would like to come in and **rap** about it.

4. If you do not mind my saying so Doctor, "Exegetical Dissertations on Theosophical Redirections in the Twentieth Century" will not have quite the impact of "Keep the Faith **Baby**".

5. Woe! The gods have now decreed that there will be a short pause for a **commercia**l.

6. And finally Lord may **thy** wisdom guide and inspire the deliberations of those thy servants our planning and zoning commission.

7. I deem thee **newsworthy**.

8. You are getting to be a big boy sonny and when we are alone it is still nice to hear you call me Popsy. But in front of people try to remember to call me **Sire**.

9. Miss would you type this searing blast at the **honkie** power structure in triplicate for me please?

10. There is a skillful juxtaposition of contrasting elements but **doggone** it I miss his oblique offhand symbolism.

11. I warned you that giving him the presidency would be a **boo** boo.

Appendix D

All the texts output by the generator

Chapters 7 and 8 refer to this appendix. As Chapter 8 explains, the generator uses 4 thresholds and different values for these thresholds were experimented with in development testing. The two sets of thresholds shown in Table D.1 yielded the best results and both sets were used in the final test.

Table D.1: Two sets of generator thresholds were used in the final test

set #1	set #2
• frequency threshold = 20 • corpus threshold = 2 • uniformity threshold = 0.55 • joke threshold = 0.35 These more restrictive thresholds yielded the highest precision scores in development testing, but yielded only 2 outputs from 203 potential seed texts.	• frequency threshold = 4 • corpus threshold = 1 • uniformity threshold = 0.5 • joke threshold = 0.35 These less restrictive thresholds yielded lower precision scores in development testing but more texts (100) were generated from the 203 potential seed texts.

D.1 Output from the more restrictive thresholds

The more restrictive thresholds (set #1 in Table D.1) generated only 2 texts from the 203 seed texts:

1. If you develop decree, never have more than ten.

2. Presidential leadership need not always cost money. Look for low and no dues options. They can be surprisingly effective.

D.2 Output from the less restrictive thresholds

The less restrictive thresholds (set #2 in Table D.1) generated following 100 texts from the 203 seed texts:

1. And so our goal on health care is, if we can get, instead of health care costs going up percent a year, it is going up at the level of inflation, maybe just slightly above inflation, we have made huge progress. And by the way, that is the single most important thing we could do in terms of reducing our arrears. That is why we did it.

2. And so our goal on health care is, if we can get, instead of health care costs going up percent a year, it is going up at the level of inflation, maybe just slightly above inflation, we have made huge progress. And by the way, that is the single most important thing we could do in terms of reducing our dues. That is why we did it.

3. And so our goal on health care is, if we can get, instead of health care costs going up percent a year, it is going up at the level of inflation, maybe just slightly above inflation, we have made huge progress. And by the way, that is the single most important thing we could do in terms of reducing our shortfall. That is why we did it.

4. And so our goal on health care is, if we can get, instead of health care costs going up percent a year, it is going up at the level of inflation, maybe just slightly above inflation, we have made huge progress. And by the way, that is the single most important thing we could do in terms of reducing our underage. That is why we did it.

5. As a nuclear power - as the only nuclear power to have used a nuclear weapon - the United States has a moral albatross to act.

6. As a nuclear power - as the only nuclear power to have used a nuclear weapon - the United States has a moral encumbrance to act.

7. As a nuclear power - as the only nuclear power to have used a nuclear weapon - the United States has a moral subjection to act.

8. Change will not come if we wait for some other person or some other time. We are the ones we have been waiting for. We are the vicissitude that we seek.

9. I can make a firm pledge, under my plan, no family making less than a year will see any form of tax increase. Not your income tax, not your payroll tax, not your capital gains taxes, not any of your tithe.

10. I do not oppose all wars. What I am opposed to is a dumb war. What I am opposed to is a pandemic war.

11. I found this national debt, doubled, wrapped in a big bow waiting for me as I stepped into the watermelon Office.

12. I just want to go through Central Park and watch folks passing by. Spend the whole day watching mortals. I miss that.

13. I mean, I do think at a certain point you have made enough pesos.

14. I said that America role would be limited; that we would not put ground troops into Libya; that we would focus our unique capabilities on the front end of the operation, and that we would transfer responsibility to our allies and comrades.

15. I said that America role would be limited; that we would not put ground troops into Libya; that we would focus our unique capabilities on the front end of the operation, and that we would transfer responsibility to our allies and confederates.

16. I said that America role would be limited; that we would not put ground troops into Libya; that we would focus our unique capabilities on the front end of the operation, and that we would transfer responsibility to our allies and consorts.

17. I said that America role would be limited; that we would not put ground troops into Libya; that we would focus our unique capabilities on the front end of the operation, and that we would transfer responsibility to our allies and pals.

18. I said that America role would be limited; that we would not put ground troops into Libya; that we would focus our unique capabilities on the front end of the operation, and that we would transfer responsibility to our allies and playmates.

19. I said that America role would be limited; that we would not put ground troops into Libya; that we would focus our unique capabilities on the front end of the operation, and that we would transfer responsibility to our allies and sidekicks.

20. Can a man take fire in his bosom, and his ensemble not be burned?

21. Can a man take fire in his bosom, and his equipment not be burned?

22. Can a man take fire in his bosom, and his gear not be burned?

23. Can a man take fire in his bosom, and his outfit not be burned?

24. Can a man take fire in his bosom, and his tout ensemble not be burned?

25. Can a man take fire in his bosom, and his vesture not be burned?

26. Can one go upon hot coals, and his pads not be burned?

27. Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the cosmos.

28. Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the terra.

29. Wisdom is before him that hath understanding; but the eyes of a fool are in the ends of the universe.

30. A foolish son is a grief to his father, and acidity to her that bare him.

31. Always remember that back-up may hate you but those who hate you do not win unless you hate them. And then you destroy yourself.

32. Always remember that redundancy may hate you but those who hate you do not win unless you hate them. And then you destroy yourself.

33. Any change is resisted because bureaucrats have a vested interest in the bedlam in which they exist.

34. Any change is resisted because bureaucrats have a vested interest in the discord in which they exist.

35. Any change is resisted because bureaucrats have a vested interest in the entropy in which they exist.

36. Any change is resisted because bureaucrats have a vested interest in the misrule in which they exist.

37. Any change is resisted because bureaucrats have a vested interest in the tumult in which they exist.

38. Any lady who is first lady likes being first duchess. I do not care what they say, they like it.

39. Any lady who is first lady likes being first gentlewoman. I do not care what they say, they like it.

40. Certainly in the next years we shall see a woman president, perhaps sooner than you think. A woman can and should be able to do any political job that a swain can do.

41. Do not get the impression that you arouse my distemper. You see, one can only be angry with those he respects.

42. Do not get the impression that you arouse my irritability. You see, one can only be angry with those he respects.

43. Do not get the impression that you arouse my petulance. You see, one can only be angry with those he respects.

44. I am not a knave.

45. I believe in the battle whether it is the battle of a campaign or the battle of this office, which is a continuing fray.

46. I believe in the battle whether it is the battle of a campaign or the battle of this office, which is a continuing scrimmage.

47. I do not know anything that builds the will to win better than competitive gaiety.

48. I gave em a sword. And they stuck it in, and they twisted it with relish. And I guess if I had been in their position, I would have done the same implement.

49. I reject the cynical view that politics is a dirty biz.

50. In the television age, the key distinction is between the candidate who can speak poetry and the one who can only speak nonfiction.

51. It is necessary for me to establish a winner image. Therefore, I have to beat VIP.

52. It is necessary for me to establish a winner image. Therefore, I have to beat personage.

53. Arguments of convenience lack righteousness and inevitably trip you up.

54. Be precise. A lack of precision is dangerous when the margin of boner is small.

55. Be precise. A lack of precision is dangerous when the margin of misdeed is small.

56. Be precise. A lack of precision is dangerous when the margin of transgression is small.

57. Be precise. A lack of precision is dangerous when the margin of trespass is small.

58. Death has a tendency to encourage a depressing view of enmity.

59. Do not necessarily avoid sharp edges. Occasionally they are necessary to conveyance.

60. Do not speak ill of your predecessors or successors. You did not walk in their sneaker.

61. First rule of politics: you can not win unless you are on the ballot. Second commandment: If you run, you may lose. And, if you tie, you do not win.

62. First rule of politics: you can not win unless you are on the ballot. Second edict: If you run, you may lose. And, if you tie, you do not win.

63. First rule of politics: you can not win unless you are on the ballot. Second ordinance: If you run, you may lose. And, if you tie, you do not win.

64. First rule of politics: you can not win unless you are on the ballot. Second precept: If you run, you may lose. And, if you tie, you do not win.

65. First rule of politics: you can not win unless you are on the ballot. Second propriety: If you run, you may lose. And, if you tie, you do not win.

66. If a prospective Presidential approach can not be explained clearly enough to be understood well, it probably has not been thought through well enough. If not well understood by the American people, it probably will not sail anyway. Send it back for further ideation.

67. If a prospective Presidential approach can not be explained clearly enough to be understood well, it probably has not been thought through well enough. If not well understood by the American people, it probably will not sail anyway. Send it back for further musing.

68. If you develop commandment, never have more than ten.

69. If you develop decree, never have more than ten.

70. If you develop edict, never have more than ten.

71. If you develop ordinance, never have more than ten.

72. If you develop precept, never have more than ten.

73. If you develop propriety, never have more than ten.

74. If you foul up, tell the President and correct it fast. Delay only amalgamation mistakes.

75. If you foul up, tell the President and correct it fast. Delay only combo mistakes.

76. In politics, every day is filled with numerous opportunities for serious boner. Enjoy it.

77. In politics, every day is filled with numerous opportunities for serious misdeed. Enjoy it.

78. In politics, every day is filled with numerous opportunities for serious trespass. Enjoy it.

79. Know that the amount of criticism you receive may correlate somewhat to the amount of clout you receive.

80. Know that the amount of criticism you receive may correlate somewhat to the amount of promo you receive.

81. Apprehend to say I do not know. If used when appropriate, it will be often.

82. Leave the President family business to him. You will have plenty to do without trying to manage the First Family. They are likely to do fine without your succor.

83. Presidential leadership need not always cost money. Look for low and no dues options. They can be surprisingly effective. The Federal Government should be the last resort, not the first. Ask if a potential program is truly a federal responsibility or whether it can better be handled privately, by voluntary organizations, or by local or state dominion.

84. The Federal Government should be the last resort, not the first. Ask if a potential program is truly a federal responsibility or whether it can better be handled privately, by voluntary organizations, or by local or state the feds.

85. The Secretary of Defense is not a super General or Admiral. His task is to exercise civilian control over the Department for the Commander in Chief and the Arcadian.

86. The Secretary of Defense is not a super General or Admiral. His task is to exercise civilian control over the Department for the Commander in Chief and the homey.

87. The Secretary of Defense is not a super General or Admiral. His task is to exercise civilian control over the Department for the Commander in Chief and the rustic.

88. Treat each federal dollar as if it was hard earned; it was by a burgher.

89. Treat each federal dollar as if it was hard earned; it was by a inhabitant.

90. Visit with your predecessors from previous Administrations. They know the ropes and can help you see around some corners. Try to make original faux pas, rather than needlessly repeating theirs.

91. When asked for your views, by the press or others, remember that what they really want to know is the CEO views.

92. When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian bridle.

93. When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian dominion.

94. When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian subjection.

95. When cutting staff at the Pentagon, do not eliminate the thin layer that assures civilian sway.

96. Your performance depends on your people. Select the best, train them and back them. When errors occur, give sharper guidance. If errors persist or if the fit feels wrong, help them move on. The country cannot afford amateur hour in the alabaster House.

97. Your performance depends on your people. Select the best, train them and back them. When errors occur, give sharper guidance. If errors persist or if the fit feels wrong, help them move on. The country cannot afford amateur hour in the blanched House.

98. Your performance depends on your people. Select the best, train them and back them. When errors occur, give sharper guidance. If errors persist or if the fit feels wrong, help them move on. The country cannot afford amateur hour in the hoary House.

99. Your performance depends on your people. Select the best, train them and back them. When errors occur, give sharper guidance. If errors persist or if the fit feels wrong, help them move on. The country cannot afford amateur hour in the pearly House.

100. Your performance depends on your people. Select the best, train them and back them. When errors occur, give sharper guidance. If errors persist or if the fit feels wrong, help them move on. The country cannot afford amateur hour in the waxen House.

Bibliography

Alexander, R. (1984). Verbal humor and variation in english: Sociolinguistic notes on a variety of jokes. *Beiträge zur Fremdsprachenvermittlung aus dem Konstanzer Sprachlehrinstitut (SLI)*, 14:53–63.

Attardo, S. (1994). *Linguistic theories of humor*. Walter de Gruyter, Berlin.

Attardo, S., Hempelmann, C., and Maio, S. D. (2002). Script oppositions and logical mechanisms: modeling incongruities and their resolutions. *Humor: International Journal of Humor Research*, 15(1):3–46.

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):293–347.

Biber, D. (1988). *Variation across speech and writing*. Cambridge University Press.

Binsted, K. (1996). *Machine humour: An implemented model of puns*. PhD thesis, University of Edinburgh.

Brooke, J., Wang, T., and Hirst, G. (2010). Automatic acquisition of lexical formality. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 90–98. Association for Computational Linguistics.

Catford, J. C. (1965). *A linguistic theory of translation: An essay in applied linguistics*, volume 8. Oxford University Press.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Edmonds, P. (1999). *Semantic representations of near-synonyms for automatic lexical choice*. PhD thesis, University of Toronto.

Fauconnier, G. and Turner, M. (1994). Conceptual projection and middle spaces. Technical report, Citeseer.

Freud, S. (1966). *Jokes and their relation to the unconscious*. Routledge & Kegan Paul, London. Translated by James Strachey. First published 1905.

Friend, T. (2002). What's So Funny? *The New Yorker*, November 11:78–93.

Gale, W. and Sampson, G. (1995). Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2.

Halliday, M. A. (1988). On the language of physical science. *Registers of written English: Situational factors and linguistic features*, pages 162–178.

Hinton, P. (2004). *Statistics explained*. Routledge, London/New York.

Jurafsky, D., Martin, J., and Kehler, A. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall.

Kamvar, S., Klein, D., and Manning, C. (2002). Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 283–290.

Katz, B. (1993). A neural resolution of the incongruity and incongruity-resolution theories of humour. *Connection Science*, 5:59–75.

Keith-Spiegel, P. (1972). Early conceptions of humor: Varieties and issues. *The psychology of humor: Theoretical perspectives and empirical issues*, pages 4–39.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koestler, A. (1964). The act of creation. *New York*, 13.

Kyratzis, S. (2003). Laughing metaphorically: metaphor and humour in discourse. In *8th International Cognitive Linguistics Conference*, pages 20–25.

Lakoff, G. and Johnson, M. (2003). Metaphors we live by. 1980. *Chicago: U of Chicago P*.

Landauer, T., Foltz, P., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25:259–284.

Lessard, G. and Levison, M. (1992). Computational modelling of linguistic humour: Tom Swifties. In *ALLC/ACH Joint Annual Conference, Oxford*, pages 175–178.

Lessard, G. and Levison, M. (1993). Computational modelling of riddle strategies. In *ALLC/ACH Joint Annual Conference, Georgetown University, Washington, DC*, pages 120–122.

Lessard, G. and Levison, M. (2005). Computational generation of limericks. *Literary and linguistic computing*, 20:89.

Li, X. and King, I. (1999). Gaussian mixture distance for information retrieval. In *Proceedings of the International Conference on Neural Networks*, pages 2544–2549.

Lundmark, C. (2003). Puns and blending: The case of print advertisements. *Paper, Luleå University of Technology, Sverige on http://wwwling. arts. kuleuven. ac. be/iclc/Papers/Lundmark. pdf*.

Manly, B. (2004). *Multivariate statistical methods: a primer*. Chapman & Hall/CRC.

Manning, C. and Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D., and Black, R. (2008). The construction of a pun generator for language skills development. *Applied Artificial Intelligence*, 22(9):841–869.

Martin, R. A. (2007). *The Psychology of Humor: an integrative approach*. Elsevier Academic Press, London.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., et al. (2011). Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Mihalcea, R. and Pulman, S. (2007). Characterizing humour: An exploration of features in humorous texts. *Lecture Notes in Computer Science*, 4394:337–347.

Mihalcea, R. and Strapparava, C. (2005). Computational laughing: Automatic recognition of humorous one-liners. In *Proc. of the 27th Annual Conference of the Cognitive Science Society*

(COGSCI 05).

Mihalcea, R. and Strapparava, C. (2006). Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence*, 22(2):126–142.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nerhardt, G. (1970). Humor and inclination to laugh: emotional reactions to stimuli of different divergence from a range of expectancy. *Scandinavian Journal of Psychology*, 11:185–195.

Oring, E. (2003). *Engaging Humor*. University of Illinois Press.

Paiva, D. (2004). *Using Stylistic Parameters to Control a Natural Language Generation System*. PhD thesis, PhD Thesis, University of Brighton, Brighton, UK.

Paiva, D. and Evans, R. (2004). A framework for stylistically controlled generation. *Natural Language Generation*, pages 120–129.

Paiva, D. and Evans, R. (2005). Empirically-based control of natural language generation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 58–65. Association for Computational Linguistics.

Partington, A. (2008). From Wodehouse to the White House: A corpus-assisted study of play, fantasy and dramatic incongruity in comic writing and laughter-talk. *Lodz Papers in Pragmatics*, 4.2:189–213.

Raskin, V. (1985). *Semantic Mechanisms of Humor*. Reidel, Dordrecht.

Ritchie, G. (1999). Developing the incongruity-resolution theory. In *Proceedings of the AISB Symposium on Creative Language: Stories and Humour*, pages 78–85.

Ritchie, G. (2004). *The linguistic analysis of jokes*. Routledge, London/New York.

Ritchie, G. (2009a). Can computers create humor? *AI Magazine*, 30(3).

Ritchie, G. (2009b). Variants of incongruity resolution. *Journal of Literary Theory*, 3(2):313–332.

Salton, G. (1971). The SMART retrieval system: experiments in automatic document processing.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Salton, G. and McGill, M. (1986). *Introduction to modern information retrieval*. McGraw-Hill, New York, NY.

Shlens, J. (2009). A tutorial on principal component analysis. *Online Note: http://www.snl.salk.edu/ shlens/pub/notes/pca.pdf*.

Stock, O. and Strapparava, C. (2003). HAHAcronym: Humorous agents for humorous acronyms. *Humor : International Journal of Humor Research*, 16(3):297–314.

Stone, P. (1997). Thematic text analysis: New agendas for analyzing text content. *Text analysis for the social sciences: Methods for drawing statistical inferences from texts and transcripts*, pages 35–54.

Suls, J. (1983). Cognitive processes in humor appreciation. *Handbook of humor research*, 1:39–57.

Suls, J. M. (1977). Cognitive and disparagement theories of humour: A theoretical and empirical synthesis. pages 41–45.

Turney, P. (2006). Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Turney, P. and Littman, M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems-TOIS*, 21(4):315–346.

Turney, P., Pantel, P., et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Veale, T. (2004). Incongruity in humor: root cause or epiphenomenon? *Humor*, 17(4):419–428.

Venour, C., Ritchie, G., and Mellish, C. (2010). Quantifying Humorous Lexical Incongruity. In Ventura, D., Pease, A., y Perez, R. P., and Ritchie, G., editors, *Proceedings of the 1st International Conference on Computational Creativity*, pages 268–277.

Venour, C., Ritchie, G., and Mellish, C. (2011). Dimensions of incongruity in register humour. In Dynel, M., editor, *The Pragmatics of Humour across Discourse Domains*, pages 125–144. John Benjamins.

Wilson, M. (1987). *MRC psycholinguistic database: machine usable dictionary; version 2.00*. Citeseer.

Zhao, Y. and Karypis, G. (2003). Criterion functions for document clustering. *Technical Report TR 01–40*.

Zhao, Y. and Karypis, G. (2004). Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331.