Recognising Affect in Text using Pointwise-Mutual Information

Jonathon Read

Submitted for the degree of Master of Science

University of Sussex

September 2004

Abstract

This dissertation describes experiments conducted to evaluate an algorithm that attempts to automatically recognise emotions (*affect*) in written language. Examples from several areas of research that can inform affect recognition experiments are reviewed, including sentiment analysis, subjectivity analysis, and the psychology of emotion. An affect annotation exercise was carried out in order to build a suitable set of test data for the experiment. An algorithm to classify according to the emotional content of sentences was derived from an existing technique for sentiment analysis. When compared against the manual annotations, the algorithm achieved an accuracy of 32.78%. Several factors indicate that the method is making slightly informed choices, and could be useful as part of a holistic approach to recognising the affect represented in text.

Acknowledgements

Thanks to John Carroll, not only for supervision and helpful advice during this project but also for lectures in Natural Language Processing. Thanks also to Diana McCarthy and the other members of NLCL for their kind welcome and discussions during the STATNLP reading group.

I greatly appreciate the provision of access to the Waterloo MultiText System by Peter Turney, without which this project would have been significantly more time-consuming.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

This dissertation describes experiments conducted to evaluate an algorithm that attempts to automatically recognise emotions (*affect*) in written language. The project included the development of a system to collect training and test data using the World Wide Web, and the adaptation of an existing algorithm of sentiment classification to affect recognition.

Affect recognition is essentially a text classification endeavour. When attempting to classify text we assign one or more classes to the text. In the case of affect recognition we assign classes that represent each type of emotion that needs to be classified. Natural language processing techniques have been applied with varying degrees of success to the classification of documents in terms of newswire (McCallum and Nigam 1998), genre (Finn et al. 2002) and sentiment (Pang et al. 2002; Turney 2002).

When considering affect classification, however, it should be clear that the boundaries between classes are far more subtle than in other text classification problems. All users of language understand that not only can an emotion be described in many ways, but also that one description can be interpreted as several different kinds of emotional states, depending on the interpreter's personal experience. The problem is further complicated by the ambiguous nature of the words that represent emotion. This dissertation approaches the former problem by including a vast range of personal experiences of many individuals using a very-large corpus. It tries to tackle the latter problem by attempting to disambiguate emotion-bearing words with regards to the type of affect that they represent.

Affect-classification algorithms should perhaps be grouped with other forms of *affective computing*, a recently-proposed area of research (Picard 1997). Affective computing seeks to equip future human-computer interfaces with an understanding of emotion and an ability to recognise affect in their users. Recognition of affect in text may aid certain interfaces, for example by warning users of potentially ambiguous affect content in electronic communications. It can also aid interaction between humans and artificial agents; Nass et al. (1994) studied human computer interactions, finding that people seem to interact most successfully with their computers in a social and affectively meaningful way.

1.1. Dissertation Focus

This dissertation and project focuses on the development of a linguistic model (based on analysing word similarity) to be used to computationally classify sentences according to their affective content. This focus necessitated a sub-project involving the construction of manually-annotated test data from the fiction domain). An existing algorithm to determine the general sentiment of text (Turney 2002), which works well when analysing product reviews, is applied to the more general domain of fiction as an interesting further experiment.

1.2. Dissertation Organisation

This dissertation is structured as follows: Chapter 2 reviews the literature that influenced my experiments, including previous work in affect recognition, and research into subjectivity and sentiment analysis, drawing from machine-learning, information retrieval, and measures of similarity between words. Two models of affect proposed by psychologists are also reviewed as to their usefulness in providing classes for a computational linguistics experiment.

I describe the considerations made and steps taken to construct a corpus of text with a heavy affective content, in Chapter 3.

The methods used to manually annotate this corpus with classifications with regards to the sentiment and affect of the sentences are detailed in Chapter 4. This chapter also explains two metrics that evaluate the annotations, including the kappa coefficient of agreement.

Chapter 5 details the algorithm used to classify affect. It discusses the choice of algorithm, selecting phrases from the text, estimating the overall affect class and optimisation of the algorithm.

The results of the experiments conducted to apply the algorithm to the test data created and annotated in previous chapters are presented in Chapter 6. This chapter also considers a variety of evidence that indicates the potential effectiveness of the algorithm.

My conclusions are presented in Chapter 7, including suggestions for future avenues of research to be explored and possible applications of reliable affect recognition technology.

Chapter 2 Literature Review

This chapter presents various potential sources of influence to an affect classification experiment.

2.1. The Psychology of Affect

An important starting point in an experiment to computationally recognise the type of affect represented in text is the selection of suitable classes of emotions. This section therefore presents two different models of affect proposed by researchers in the field of Psychology and considered in detail for use in these experiments. Other models include Basic Emotion, with classes of anger, disgust, fear, joy, sadness and surprise (Ekman 1992), which is based on facial expressions of emotion, and Psychoevolutionary Theory, with classes of acceptance, anger, anticipation, disgust, joy, fear, sadness and surprise (Plutchik 1980), which is based on their relations to biological processes.

The first model considered in detail for this experiment is the Cognitive Structure of Emotions (Ortony et al. 1988), and is depicted in Figure 2.1. The authors base their model on the premise that an emotion is a valenced reaction to either: events (pleased versus displeased), agents (approving versus disapproving) or objects (liking versus disliking).

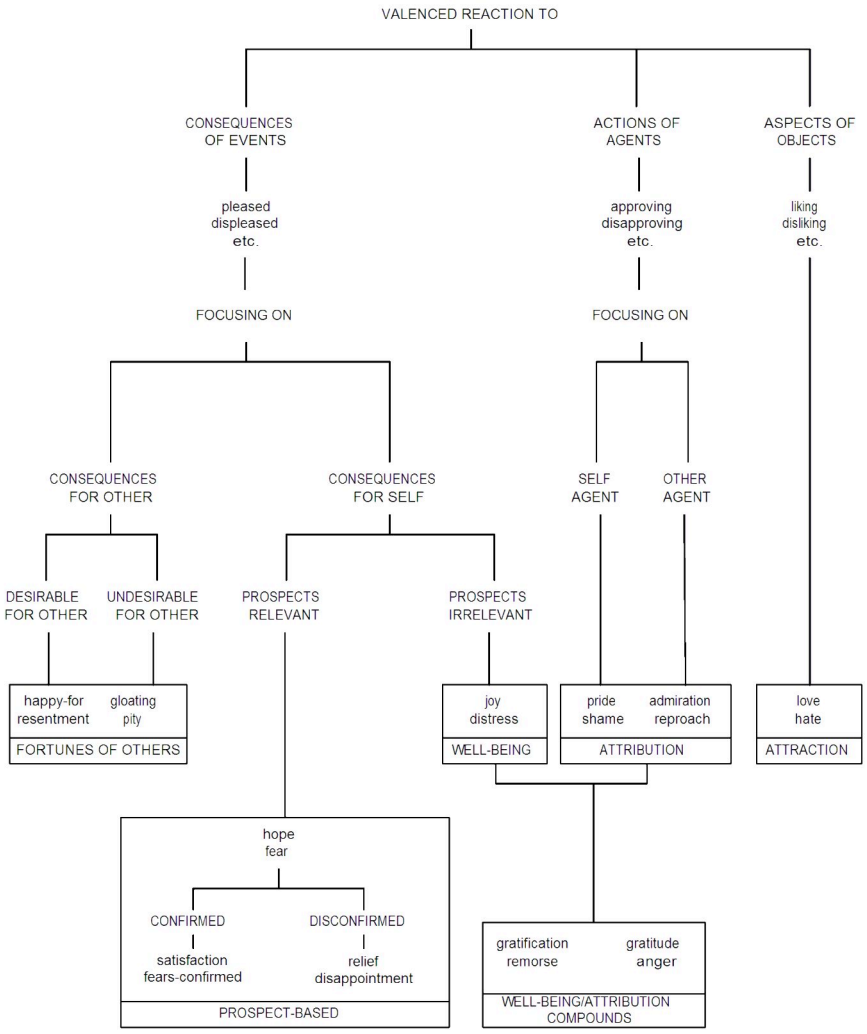

Figure 2.1: The Cognitive Structure of Emotions (Ortony et al. 1988)

Reactions to events break into three groups – those concerning the fortunes of others (happy-for, resentment, gloating and pity), those concerning prospects (satisfaction, fears-confirmed, relief and disappointment), and those concerning general well-being (joy and distress). Reactions to agents include those to oneself (pride and shame) and those to other agents (admiration and reproach). Reactions to objects are simply varying degrees of love or hate. The well-being and attribution groups are compounded to derive a further set of classes (gratification, remorse, gratitude and anger).

The authors describe numerous variables that influence the strength of the valenced reaction (desirability for event-based emotions, praiseworthiness for agent-based emotions and appealingness for object-based emotions, among many others).

The authors assert that the strength of each emotion experienced will be different (depending on the variables they define). Further information that might improve the model's usefulness in a computational linguistics experiment could be a scale of words associated with the strength of the emotions. This would provide a set of synonyms through to antonyms and a rough estimate of the strength of association. For example, one might suspect that reactions to objects might scale from *hate* to *dislike* to *indifferent* to *like* to *love*.

A somewhat less-complex alternative to the cognitive structure of emotions is a Two-Factor Structure of Affect (Watson and Tellegen 1985). The authors of this model describe emotions in only two dimensions; that of Positive Affect and Negative Affect. Positive Affect scales between High (energised states with a positive association, such as feeling elated) and Low (states associated with sleepiness). Negative Affect scales between High (energised states with a negative association, such as fear) and Low (states associated with inactivity and a lack of emotion).

The authors also define two further dimensions of Pleasantness and Engagement. They state that these dimensions are derived from a combination of Positive Affect and Negative Affect, as depicted schematically in Figure 2.2. Thus, for example, Pleasantness is a compound of High Positive Affect and Low Negative Affect while Disengagement is a compound of Low Positive Affect and Low Negative Affect.

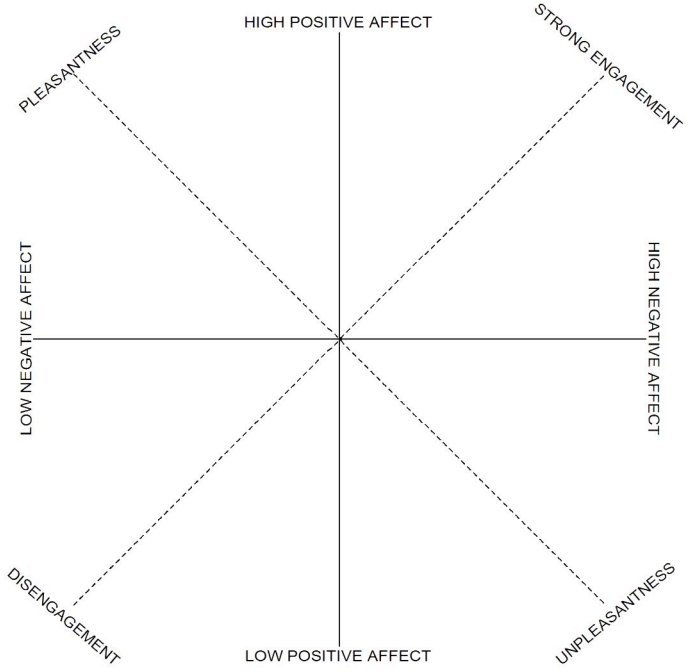

Figure 2.2: The Two-Factor Structure of Affect (Watson and Tellegen 1985)

The authors provide mood words (as reproduced in Table 2.1) that have been qualitatively ascribed to the affect classes. However, these words are not given a scalar value indicating where they occur on the respective axes of the model.

Dimension	High-pole	Low-pole
Positive Affect	active, elated, enthusiastic, excited, peppy, strong	drowsy, dull, sleepy, sluggish
Negative Affect	distressed, fearful, hostile, jittery, nervous, scornful	at rest, calm, placid, relaxed
Pleasantness	content, happy, kindly, pleased, satisfied, warm-hearted	blue, grouchy, lonely, sad, sorry, unhappy
Engagement	aroused, astonished, surprised	quiescent, quiet, still

Table 2.1: Mood Words Describing Dimensions of Affect (Watson and Tellegen 1985)

2.2. Affect Recognition

Research in the area of affect recognition in text is currently rooted in the exploitation of human-supplied knowledge of emotion. Since the nature of affect is inherently ambiguous (both in terms of the affect classes and the natural language words that represent them), some researchers have elected to use fuzzy logic to represent this ambiguity (Subasic and Huettner 2001). The authors developed an affect lexicon using the value-judgements of a single linguist. This is clearly a highly subjective process, so the authors hope to extend their lexicon in the future using the contributions of multiple linguists. Each entry in the affect lexicon is comprised of five tuples; (1) the word itself, (2) the part of speech, (3) the class of affect, (4) the *centrality* – the degree to which the word is related to the class and (5) the *intensity* – the strength to which the word represents the affect class. There will typically be multiple entries for each word, differing in terms of part of speech and in affect class.

Development of the affect lexicon has been continued by other researchers (Grefenstette et al 2004). These researchers have utilised Turney's SO-PMI-IR method (see the section below on Sentiment Analysis) in order to validate the existing lexicon and automatically mine the World Wide Web to determine new entries.

The affect lexicon is then used to derive a fuzzy thesaurus, containing entries describing the degree to which affect classes are related. The values supplied by the linguist are also used to cluster classes together, to form affect category groups.

To determine the affective content of a text it is first parsed and cross-referenced against the affect lexicon. Matches of word and part of speech are used to build the affect set that represents the text. The centralities and intensities of the text's affect set are combined using fuzzy logic resulting in a list of classes and the degree to which each class is central to the text.

This approach is useful as it recognises that the affect that is represented by a piece of text is a rich tapestry of layers. However, at this stage it suffers from several drawbacks. Firstly, it is rooted in the subjective interpretations of a single person. The nature of affect is highly personal and humans will never agree on the details of an affect lexicon. Consequently humans may not agree on the output of a system based on such a lexicon. Secondly, the approach is also clearly limited by the persistent problem of knowledge acquisition, relying as it does on an expert-built affect lexicon. Finally, the structure of the text is ignored. The authors do not appear to make any accommodations for the way closed-class words, word order or grammatical structure can change the meaning of a sentence (for example, if interpreting the sentence *'He was not angry'* the system will ignore the negation and simply extract the adjective *'angry'*).

A useful overview of these 'contextual valence shifters' has been presented recently (Polanyi and Zaenen 2004). In their paper the authors consider how the organisation of text can impact the class of affect being represented. These include negations, intensifiers, modal operators, presuppositions and connectors.

2.3. Sentiment Analysis

While research in affect recognition is still relatively sparse, there are several approaches being developed that classify the semantic orientation, with regards to the general sentiment, of text. That is, recognising if the author's opinion towards his or her subject is generally positive or generally negative.

Three such models that have proved successful in other areas of text classification are Naïve Bayes, Maximum Entropy and Support Vector Machines (Pang et al. 2002). Each model requires a corpus of training data (in this case two sets of texts – classified by human coders into positive and negative). N-gram features are extracted from the test document, and the observed probabilities of each feature are used to determine the sentiment class.

The models performed consistently well when tested in the domain of movie reviews, ranging from 78.7% accuracy for Naïve Bayes to 82.9% accuracy for Support Vector Machines, compared to a baseline of 50%. Further work has investigated how accuracy can be improved by disregarding sentences that are objective (Pang and Lee 2004). While these results are encouraging, it is important to remember that the models' accuracy is dependent on a reasonably large set of training data. The experiments are also domain-biased as the training data is from the same domain as their test data; it is unclear how well a model trained on movie reviews might perform on a different domain.

An alternative method of semantic orientation analysis which does not require a large body of training text is Semantic Orientation using Pointwise Mutual Information and Information Retrieval, or SO-PMI-IR (Turney 2002). In this approach phrases containing adjectives or adverbs are extracted from the test document. The pointwise mutual information (PMI) of each phrase and two paradigm word sets are calculated; the average PMI of all the phrases is used to estimate the semantic orientation. The paradigm words describe the opposing poles of semantic orientation (that is, *positive* versus *negative*) and are reproduced in Table 2.2.

Positive	Negative
good, nice, excellent, positive, fortunate, correct, superior	bad, nasty, poor, negative, unfortunate, wrong, inferior

Table 2.2: Paradigm Words of Semantic Orientation (Turney and Littman 2002)

The Pointwise Mutual Information (that is, the statistical dependency between the phrase and each pole of semantic orientation), is estimated using a very large corpus, the World Wide Web as seen through a search engine, for example. The search engine is issued with queries to determine the number of cooccurences with the paradigm word sets and these hit counts are used as inputs to the function shown below.

$$\mathrm{SO}(phrase) = \log_2\left(\frac{\mathrm{hits}(phrase\ \mathrm{NEAR}\ positive)\,\mathrm{hits}(negative)}{\mathrm{hits}(phrase\ \mathrm{NEAR}\ negative)\,\mathrm{hits}(positive)}\right)$$

If the resultant average PMI is greater than zero then the review is estimated to be positive, else it is thought to be negative.

SO-PMI-IR was tested in a variety of review domains achieving an overall accuracy of 74.4%. In the domain of movie reviews, however, it was noticeably poorer than the other machine learning methods achieving 65.8% accuracy. The model also suffers in terms of speed due to the large volume of queries that need to be made to the search engine. Despite these disadvantages it is still a very useful method as it does not require training and can be easily transferred to different domains.

Interestingly, SO-PMI-IR and Naïve Bayes are far from dissimilar. Beineke et al. (2004) published formal proof that SO-PMI-IR effectively generates a virtual corpus of labelled documents and applies a Naïve Bayes classifier. SO-PMI-IR is a pseudo-supervised

approach by virtue of the fact that it operates from a small amount of supplied knowledge about words that represent the poles of semantic orientation.

2.4. Subjectivity Analysis

Another related field of research is that of subjective language recognition. Riloff and Wiebe (2003) describe a bootstrapping process for learning subjective expressions. They hypothesised that extraction patterns can represent subjective expressions that have noncompositional meanings. For example, *<x> drives <y> up the wall*, where *x* and *y* are arbitrary noun phrases. These patterns would be able to extract a wider variety of phrases than would be possible using n-grams.

The first part of their process uses a large collection of unannotated text and two high precision classifiers of semantic orientation (one for positive and one for negative). These classifiers use subjectivity clues to label sentences as positive or negative only if confidence is high; otherwise they are left unlabeled. The labelled sentences are then fed to an extraction pattern learner, which creates a set of extraction patterns that are statistically correlated with the subjective statements. These patterns are then used to supplement the subjectivity clues in the first part of the algorithm. The entire process is then bootstrapped.

The precision of their process ranged from 71% to 85% with the customary trade-off between precision and recall.

Another approach is a semi-automatic method of extracting 3-tuple representations of opinion by determining cooccurrence patterns (Kobayashi et al. 2004). The tuple describes the Subject, Attribute and Value of an opinion. The iterative process generates candidate cooccurrence patterns which are then selected to populate dictionaries of subjects, attributes and values. Candidates are generated using a selection of web documents, a set of human-supplied cooccurrence patterns, and the latest versions of the subject, attribute and value dictionaries.

When compared with the results of manually collected expressions, the coverage did not seem sufficient at around 40%. The authors ascribe this to the method not looking beyond its limited cooccurrence patterns, and hierarchical attributes (for example “the sound of the engine”).

2.5. Web-based Data Annotation

The World-Wide-Web is frequently being used to collect large quantities of data for various applications such as training data for machine-learning models and test data for all types of algorithms. One such existing example is WebExp, which provides a set of Java classes for use in implementing online psychological questionnaires. The software offers two paradigms for experiments – magnitude estimation and sentence completion – and provides facilities to randomise the materials used and carry out basic reliability tests on data provided by participants (WebExp 2004).

Another web-based project, The OpenMind Initiative, seeks to develop a database of common sense knowledge. The knowledge is acquired from three different sources (Stork 2000): (1) *domain experts* supply the fundamental algorithms that support the learning process; (2) *infrastructure developers* build the web-based interfaces that acquire the knowledge; and (3) *non-experts* contribute the raw data using these interfaces. The OpenMind framework includes tests to determine the reliability of the raw data in an attempt to only accept data that is of high quality and consistency.

The following chapter describes the steps taken to build a corpus to be used to test an affect recognition algorithm.

Chapter 3 Corpus Construction

This chapter discusses the steps taken to find a corpus of text suitable for an affect and sentiment recognition experiment. A number of existing corpora were considered for these experiments, perhaps the most suitable being the MPQA Corpus of Opinion Annotations (Wiebe 2004), which contains 530 news articles that have been manually annotated for their opinion content. While subjective language is clearly present in this corpus, it appeared that the affective content was rather sparse.

3.1. Corpus Source

It was therefore decided to build a new set of test data – one that was likely to contain a larger proportion of affective language. It seems intuitively that the domain of fiction would be most likely to satisfy this need, and the test data was built from a set of stories on a website called Fifty Word Fiction. This source was selected as each story is fifty words long – potentially compelling the authors to use highly emotional language. XML was used to encode the corpus as it is well-adopted and supported by basic processing software (including Java libraries). It is also easily re-used due to the emerging technology of XSL which provides the facility to transform XML documents (Carletta et al. 2002).

Some 155 stories were downloaded and stripped of any HTML mark-up and superfluous information. The RASP system (see below) was then used to tokenise, part-of-speech tag and lemmatise raw text files. The tokenised text files were then organised into a single XML document, with part-of-speech tags and lemmatised forms included as attributes of the tokens. An example document is listed in figure 3.1, with outputs from RASP in figures 3.2 (tokenised), 3.3 (tagged) and 3.4 (lemmatised). Figure 3.5 shows the document in the XML format. The schema for the final document is shown in Appendix A. The schema also includes provision for the addition of attributes that describe the sentiment and affect of the sentences, and a measurement of the usefulness of the associated annotations. These attributes are populated from the Affect Annotation Experiment (see Chapter 4).

The program responsible for building the corpus is listed in Appendix C as `uk.ac.sussex.jlr24.fwf.CorpusToXML`. The completed corpus was named the Fifty Word Fiction Corpus (FWF Corpus).

```
He left because it was boring.  From village to city with no money, luckily he met Kelly. She was loaded so
they got together.  He didn't ask why she had money though, so he was soon sorry when she disappeared.  He
was left, tied up, dying in a burning nightclub.
```

Figure 3.1: Example Document from Fifty-Word Fiction

```
^ He left because it was boring . ^ From village to city with no money , luckily he met Kelly . ^ She was
loaded so they got together . ^ He did n't ask why she had money though , so he was soon sorry when she
disappeared . ^ He was left , tied up , dying in a burning nightclub.
```

Figure 3.2: Tokenised Document from RASP

```
^_^
He_PPHS1
left_VVD
because_CS
it_PPH1
was_VBDZ
boring_JJ
._.
^_^
From_II
village_NN1
to_II
city_NN1
with_IW
no_AT
money_NN1
,_,
luckily_RR
he_PPHS1
met_VVD
Kelly_NP1
._.
^_^
She_PPHS1
was_VBDZ
loaded_VVN
so_RR
they_PPHS2
got_VVD
together_RR
._.
^_^
He_PPHS1
did_VDD
n't_XX
ask_VV0
why_RRQ
she_PPHS1
had_VHD
money_NN1
though_CS
,_,
so_RR
he_PPHS1
was_VBDZ
soon_RR
sorry_JJ
when_CS
she_PPHS1
disappeared_VVD
._.
^_^
He_PPHS1
was_VBDZ
left_VVN
,_,
tied_VVD
up_RP
,_,
dying_VVG
in_II
a_AT1
burning_JJ
nightclub_NNSB1
._.
```

Figure 3.3: PoS Tagged Document from RASP

```
^_^
he_PPHS1
leave_VVD
because_CS
it_PPH1
be_VBDZ
boring_JJ
._.
^_^
from_II
village_NN1
to_II
city_NN1
with_IW
no_AT
money_NN1
,_,
luckily_RR
he_PPHS1
meet_VVD
Kelly_NP1
._.
^_^
she_PPHS1
be_VBDZ
load_VVN
so_RR
they_PPHS2
get_VVD
together_RR
._.
^_^
he_PPHS1
do_VDD
not_XX
ask_VV0
why_RRQ
she_PPHS1
have_VHD
money_NN1
though_CS
,_,
so_RR
he_PPHS1
be_VBDZ
soon_RR
sorry_JJ
when_CS
she_PPHS1
disappear_VVD
._.
^_^
he_PPHS1
be_VBDZ
leave_VVN
,_,
tie_VVD
up_RP
,_,
die_VVG
in_II
a_AT1
burning_JJ
nightclub_NNSB1
._.
```

Figure 3.4: Lemmatised Document from RASP

```
?xml version="1.0" encoding="UTF-8"?>
<corpus>
    <text id="/opt/fwf-example/Docs/001">
        <sentence id="0">
            <token tag="PPHS1" lemma="he">He</token>
            <token tag="VVD" lemma="leave">left</token>
            <token tag="CS" lemma="because">because</token>
            <token tag="PPH1" lemma="it">it</token>
            <token tag="VBDZ" lemma="be">was</token>
            <token tag="JJ" lemma="boring">boring</token>
            <token tag="." lemma=".">.</token>
        </sentence>
        <sentence id="1">
            <token tag="II" lemma="from">From</token>
            <token tag="NN1" lemma="village">village</token>
            <token tag="II" lemma="to">to</token>
            <token tag="NN1" lemma="city">city</token>
            <token tag="IW" lemma="with">with</token>
            <token tag="AT" lemma="no">no</token>
            <token tag="NN1" lemma="money">money</token>
            <token tag="," lemma=",">,</token>
            <token tag="RR" lemma="luckily">luckily</token>
            <token tag="PPHS1" lemma="he">he</token>
            <token tag="VVD" lemma="meet">met</token>
            <token tag="NP1" lemma="Kelly">Kelly</token>
            <token tag="." lemma=".">.</token>
        </sentence>
        <sentence id="2">
            <token tag="PPHS1" lemma="she">She</token>
            <token tag="VBDZ" lemma="be">was</token>
            <token tag="VVN" lemma="load">loaded</token>
            <token tag="RR" lemma="so">so</token>
            <token tag="PPHS2" lemma="they">they</token>
            <token tag="VVD" lemma="get">got</token>
            <token tag="RR" lemma="together">together</token>
            <token tag="." lemma=".">.</token>
        </sentence>
        <sentence id="3">
            <token tag="PPHS1" lemma="he">He</token>
            <token tag="VDD" lemma="do">did</token>
            <token tag="XX" lemma="not">n't</token>
            <token tag="VV0" lemma="ask">ask</token>
            <token tag="RRQ" lemma="why">why</token>
            <token tag="PPHS1" lemma="she">she</token>
            <token tag="VHD" lemma="have">had</token>
            <token tag="NN1" lemma="money">money</token>
            <token tag="CS" lemma="though">though</token>
            <token tag="," lemma=",">,</token>
            <token tag="RR" lemma="so">so</token>
            <token tag="PPHS1" lemma="he">he</token>
            <token tag="VBDZ" lemma="be">was</token>
            <token tag="RR" lemma="soon">soon</token>
            <token tag="JJ" lemma="sorry">sorry</token>
            <token tag="CS" lemma="when">when</token>
            <token tag="PPHS1" lemma="she">she</token>
            <token tag="VVD" lemma="disappear">disappeared</token>
            <token tag="." lemma=".">.</token>
        </sentence>
        <sentence id="4">
            <token tag="PPHS1" lemma="he">He</token>
            <token tag="VBDZ" lemma="be">was</token>
            <token tag="VVN" lemma="leave">left</token>
            <token tag="," lemma=",">,</token>
            <token tag="VVD" lemma="tie">tied</token>
            <token tag="RP" lemma="up">up</token>
            <token tag="," lemma=",">,</token>
            <token tag="VVG" lemma="die">dying</token>
            <token tag="II" lemma="in">in</token>
            <token tag="AT1" lemma="a">a</token>
            <token tag="JJ" lemma="burning">burning</token>
            <token tag="NNSB1" lemma="nightclub">nightclub</token>
            <token tag="." lemma=".">.</token>
        </sentence>
    </text>
</corpus>
```

Figure 3.5: Document in XML Form

3.2. RASP System

RASP is a robust, accurate, domain-independent statistical parser (Briscoe and Carroll 2002) comprised of five distinct modules, as depicted in Figure 3.6. The output from each module is a text file listing the tokens and attributes determined by the module, the final output being sets of grammatical relations for each sentence in the text.

I created a package of classes to form an interface between Java and the RASP system to expedite the development of the FWF Corpus. The creation of this corpus only called for the Tokeniser, PoS Tagger and Lemmatiser modules however, so the Parser and Grammar, and Statistical Disambiguator modules are not used at this time. The interface simply calls an external process – in this case a UNIX shell script which in turn invokes each of the required RASP system modules. The Java interface then reads the text files generated by RASP building its own internal representation, before deleting the text files. The classes that make up the Java-RASP Interface are held in the package `uk.ac.sussex.jlr24.rasp`, and are listed in Appendix B.

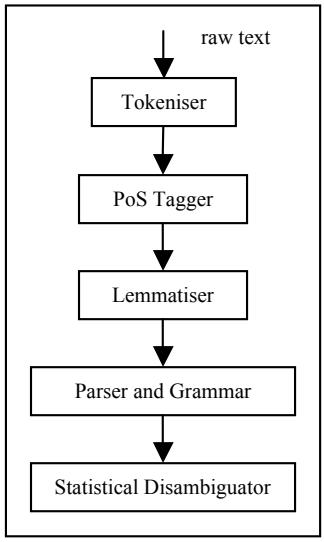

Figure 3.6: RASP System Architecture (Briscoe and Carroll 2002)

The next chapter discusses an experiment undertaken to annotate the sentences of FWF Corpus with classifications of sentiment and affect.

Chapter 4 Affect Annotation Experiment

This chapter describes an experiment to create a set of data that could potentially be used to train or test an affect and sentiment recognition algorithm. An online experiment was carried out in order to manually annotate each of the 758 sentences in the FWF Corpus with classifications in terms of its sentiment and affect. The classes chosen for sentiment were that of *positive*, *negative* and *unclassifiable.* The Two-Factor Structure of Affect (Watson and Tellegen 1985) axis labels – *high positive affect, low positive affect, high negative affect, low negative affect, high pleasantness, low pleasantness* (unpleasantness)*, high engagement, low engagement* (disengagement) and *unclassifiable* – were chosen as the affect classes. An additional output was a numerical value describing the degree of confidence in the manual annotation.

A small website was developed utilising JServlets to record the annotations made. The source code for this website is listed in Appendix C - see `AnnotationStartServlet` and `AnnotationServlet`. Annotators were presented with a page (depicted in Figure 4.1) containing the sentence to annotate and two sets of radio buttons; one for sentiment choice and the other for affect choice. Since the class names seemed rather unclear to those unfamiliar with the Two-Factor Structure of Affect, I deemed it more user friendly to use the mood word indicators listed by Watson and Tellegen for the affect radio button labels.

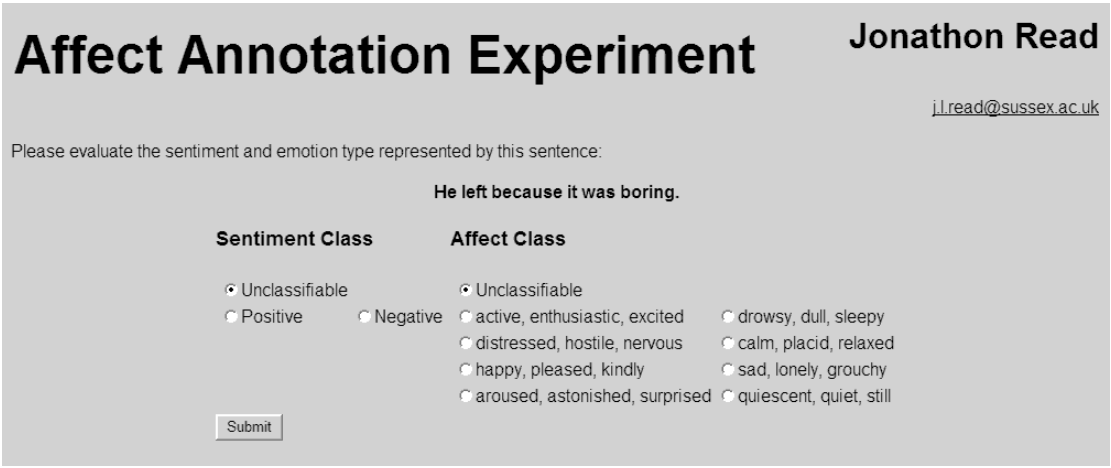

Figure 4.1: Affect Annotation Experiment Screenshot

One mistake was made in term of user-friendliness. While it was the assumption that people would simply stop of their own accord, many annotators reported being frustrated by the endless number of annotations they were asked to perform. The annotations could possibly be made more reliable in future experiments by limiting the coder's sessions and asking them to come back to perform further sessions at a later date.

The experiment was online and available to the general public for one month, during which some 3,301 annotations were made by 49 annotators. The 'Initial Overall' row in tables 4.1 and 4.2 show the distribution of annotations made for sentiment and affect, respectively. The other rows of the table describe the distribution of annotations at later stages of annotation analysis.

	# Annotations	Unclassifiable	Positive	Negative
Initial Overall	3301	52.56%	18.15%	29.29%
Thresholded Overall	2713	58.20%	15.44%	26.35%
Initial Expert	758	65.35%	11.33%	23.32%
Thresholded Expert	758	66.67%	10.80%	22.53%

Table 4.1: Sentiment Annotations Distribution

	# Annotations	Unclassifiable	High Positive Affect	Low Positive Affect	High Negative Affect	Low Negative Affect	High Pleasantness	Low Pleasantness	High Engagement	Low Engagement
Initial Overall	3301	47.50%	8.57%	1.88%	16.15%	3.79%	6.57%	7.18%	5.21%	3.15%
Thresholded Overall	2713	54.04%	7.45%	1.40%	15.15%	3.17%	5.93%	6.01%	4.61%	2.25%
Initial Expert	758	71.41%	4.87%	0.92%	12.25%	0.79%	3.16%	3.69%	2.11%	0.79%
Thresholded Expert	758	73.78%	4.87%	1.05%	11.99%	0.79%	2.24%	3.29%	1.58%	0.40%

Table 4.2: Affect Annotations Distribution

The irregular class of *unclassifiable* is clearly the most common among both sentiment and affect annotations. It is interesting how *negative* (sentiment) and *high negative affect* (affect) account for the largest proportions of the regular classes. This could be because human beings are more highly attuned to identifying negative emotions, that is, situations that might threaten their well-being.

The sections that follow describe analysis that was carried out against the annotations made, starting with the following definitions:

Classes

$$\mathbf{C}_{sentiment} = \{S_u, S_p, S_n\}$$

$$\mathbf{C}_{affect} = \{A_u, A_{hpa}, A_{lpa}, A_{hna}, A_{lna}, A_{hpl}, A_{lpl}, A_{hen}, A_{len}\}$$

Number of sentences

$$s = 758$$

An annotator

$$\mathbf{A}_s = \text{choice}(c \in \mathbf{C})$$

All annotators

$$\mathbf{X}_{\text{number of annotators}, \mathbf{A}}$$

Corpus

$$\mathbf{Y}_{s, c \in \mathbf{C}} = \underset{\mathbf{A} \in \mathbf{X}}{\text{count}}(\mathbf{A}, c)$$

4.1. Expert Coder

As in the work of Passonneau and Litman (1993), an 'expert coder' was derived from the human coders' annotations, by assuming that the majority is always right. For each sentence the classification with the highest number of annotations is assumed to be the correct decision. If two or more classes tie as the most highly annotated class, then the expert coder's annotation is taken to be *unclassifiable.*

The expert coder's annotation distribution is shown in the row 'Initial Expert' in tables 4.1 and 4.2 for sentiment and affect respectively.

Expert Coder

$$\mathbf{E}_i = \max_{c \in \mathbf{C}} \arg \mathbf{X}_{i,c}$$

4.2. Inter-coder Agreement

It is informative to assess the usefulness of each individual coder's annotations. The kappa coefficient of agreement is a statistic adopted by the Computational Linguistics community as a standard measure for this purpose (Carletta 1996). The kappa statistic can be used to measure the pairwise agreement between each annotator and the expert coder. Essentially it describes the proportion of agreements when corrected for the agreement that would be expected purely by chance, and is defined as below.

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

$P(A)$, the proportion of agreements, is calculated by counting the number of times the coder has made the same annotation as the expert coder and dividing that number by the total number of annotations made by the coder.

$$P(A) = \frac{\underset{a \in A}{\text{count}}(a = E_a)}{\underset{a \in A}{\text{count}}(a)}$$

$P(E)$, the probability that the coder and the expert coder would agree by chance is estimated by first determining the probability distribution of both the coder's and expert coder's choices. This is simply the proportion of annotations made to a single class and the total number of annotations made by the coder, for each class.

$$\text{dist}(A) = \frac{\underset{a \in A}{\text{count}}(a = c)}{\underset{a \in A}{\text{count}}}$$

Then, the expected chance of agreement between the given coder and the expert coder is:

$$P(E) = \sum_{c \in C} \text{dist}(A)_c \, \text{dist}(E)_c$$

When calculated for all the annotators (see `AnnotationsAnalysis` in Appendix C), the mean kappa values were 0.54 (standard deviation of 0.22) for sentiment and 0.33 (standard deviation of 0.18) for affect. Annotators were then removed from the experiment if their kappa value fell below a threshold of 0.4 as this removed annotations that were unreliable while retaining a reasonably high number of annotations. This improved the mean kappa values to 0.65 (standard deviation of 0.14) for sentiment and 0.43 (standard deviation of 0.16) for affect. The recalculated distribution of annotations is shown in tables 4.1 and 4.2, in the rows entitled 'Thresholded Overall'. The rows entitled 'Thresholded Expert' show the distribution of the expert coder's annotations after thresholding. The stacked bar graphs of figures 4.2 and 4.3 (sentiment and affect, respectively) show the change in distribution of annotations for each stage of the process.

Figure 4.4 is a scatter graph showing the sentiment and affect kappa values (both initial and thresholded) for all annotators. It demonstrates that, generally speaking, coders who exhibit higher agreement for sentiment annotations also tend to show higher agreement for affect annotations. The outliers tend to be coders who only performed a small number of annotations.

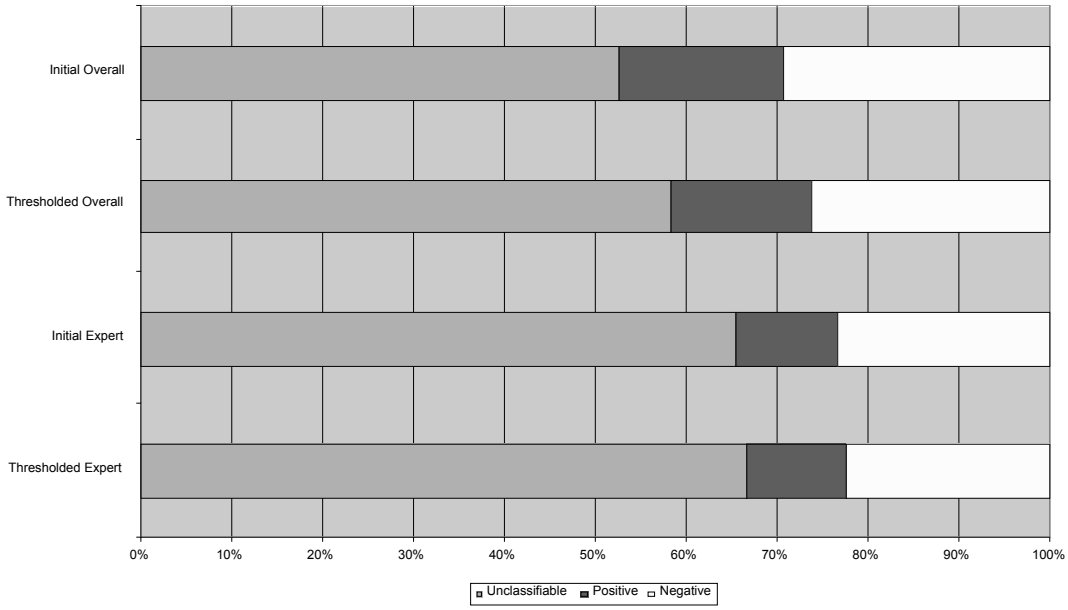

Figure 4.2: Sentiment Annotation Distributions

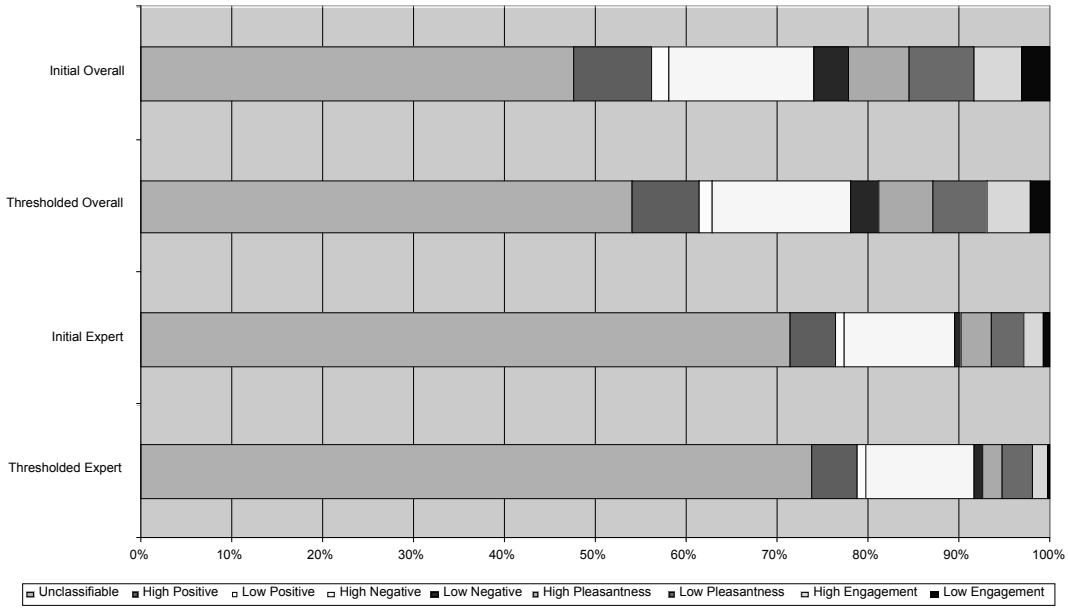

Figure 4.3: Affect Annotations Distribution

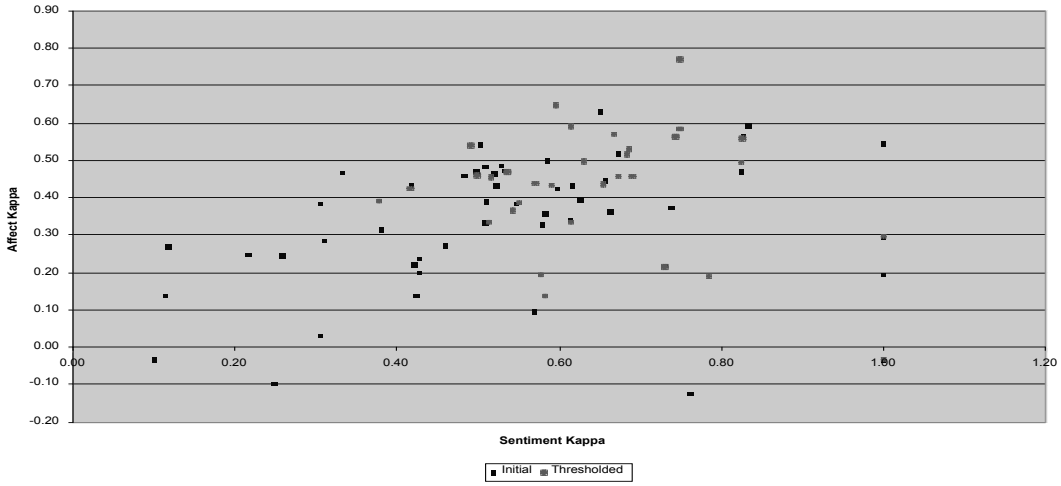

Figure 4.4: Sentiment versus Affect Annotator Agreement

4.3. Sentence Usefulness

The annotation experiment was rather informal in structure, in order to allow coders to complete as many or as few annotations as they choose to do (it was clearly unfeasible to expect all coders to complete 758 annotations). As a result, not all sentences were annotated the same number of times. It would be appropriate to assess the annotations of each sentence in order to determine their collective usefulness. Such a metric should take into account both the number of annotations made to the sentence, and the number of agreements that are observed. For this purpose I derived an n-wise measurement of agreement for a single annotative unit from the Kappa coefficient of agreement, termed the Naïve N-wise Kappa coefficient ($K_{n\text{-}wise}$).

This statistic is based upon two fundamentals of annotator agreement measurement (Carletta 1996). The first expresses the purpose of agreement analysis – to determine the degree of noise in a set of annotations. This can tell us if the data is useful for the purposes for which it has been collected. Secondly, and consequently, the expected chance of agreement must be included in any agreement measurement.

$K_{n\text{-}wise}$ is essentially the original kappa measurement, but applied to n annotations made to a single annotative unit. A weighting based on the number of agreed annotations is also applied to indicate that sentences with larger numbers of annotations are potentially more useful.

Number of classes

$$s = \text{size}(\mathbf{C})$$

Total number of annotations made to sentence

$$n = \sum_{c \in C} \mathbf{Y}_{i,c}$$

Number of annotations made to the most annotated class

$$a = \max_{c \in C} c$$

$$K_{n-wise} = a\left(\frac{\frac{a}{n} - \left(\frac{1}{s}\right)^n}{1 - \left(\frac{1}{s}\right)^n}\right)$$

This version of $K_{n\text{-}wise}$ is called naïve because it estimates the probability of agreement by chance based on the assumption that coders are equally likely to choose each of the classifications. A more complete version would make a better estimate of this by finding the product of the individual annotators' probability of choosing the chosen classification. Despite its naivety, $K_{n\text{-}wise}$ is still a useful metric, as qualitatively demonstrated by the examples shown in Table 4.3.

a	n	s	$K_{n\text{-}wise}$
1	1	3	1
2	2	3	2
0	4	3	0
2	4	3	0.988
3	4	3	2.241

Table 4.3: Example Outputs of the $K_{n\text{-}wise}$ Metric

When calculated (see `AnnotationsUsefulness` in Appendix C), the $K_{n\text{-}wise}$ value for most annotations falls between 1 and 2 and the highest between 5 and 6. This distribution of annotations according to their usefulness is shown in figures 4.5 (sentiment) and 4.6 (affect). As one might expect, the number of annotations in a division tends to decrease as the usefulness increases. In addition to being the most numerous, the classifications of *unclassifiable* and *high negative affect* seem to be those with which we can have the most confidence.

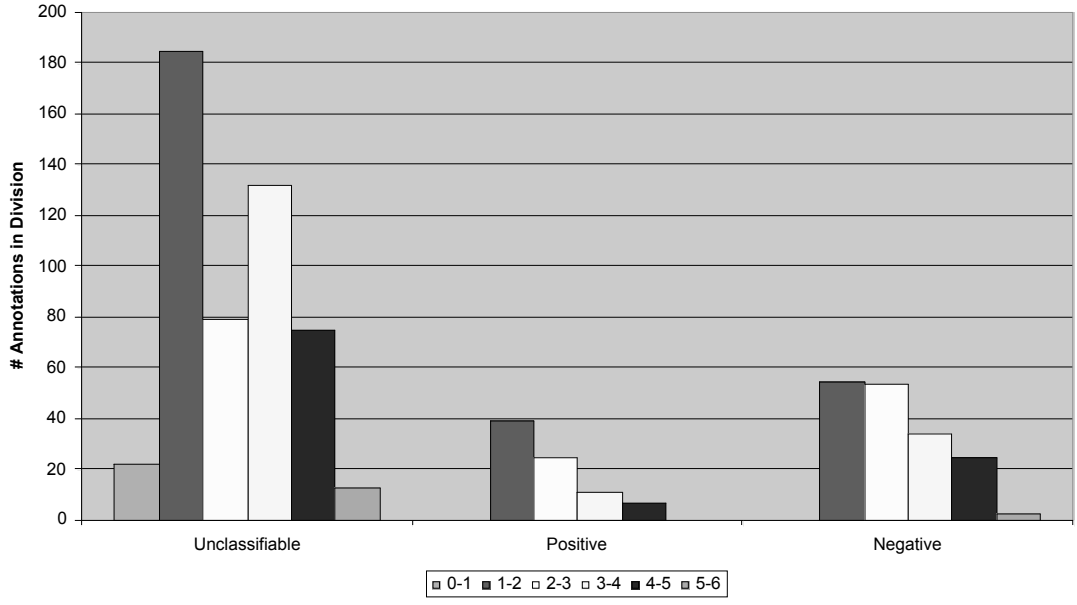

Figure 4.5: Sentiment Annotations Usefulness

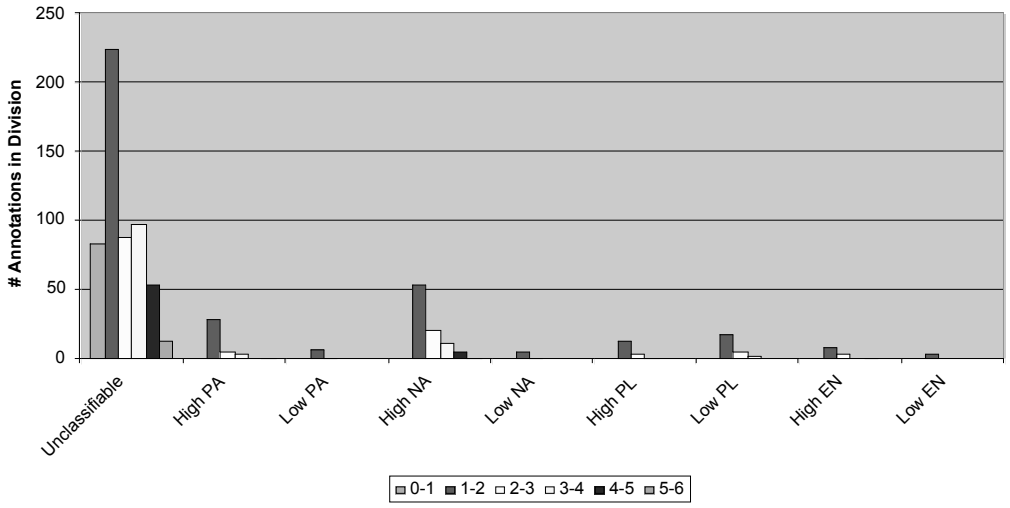

Figure 4.6: Affect Annotations Usefulness

These divisions will be used in the final experiment to test the hypothesis that sentences which humans have easily agreed upon will be easier for the computer program to classify. I expect the program's accuracy to increase with the annotation usefulness division.

The next chapter discusses the algorithms used to computationally recognise the sentiment and affect of text. These algorithms will be applied to the sentences of the FWF Corpus, and the annotations made in this experiment will be used to verify the classifications made.

Chapter 5 Affect Recognition Experiment

This chapter describes the steps that were taken to develop a program that might recognise the affect and sentiment of the sentences in the FWF Corpus. As described in Chapter 4, each sentence has been manually annotated with a sentiment (*positive*, *negative* or *unclassifiable*) and an affect (taken from Watson and Tellegen's model (1985) – *high positive affect*, *low positive affect*, *high negative affect*, *low negative affect*, *high pleasantness*, *low pleasantness*, *high engagement*, *low engagement* and *unclassifiable*). The code developed to implement the process described is listed in Appendix D.

5.1. Choice of Algorithm

A cursory glance at Table 4.2 quickly tells us that there is a sparse data problem for many of the affect classes – most notably that of *low engagement*, which accounts for just 0.4% (3 of 758) annotations. It is clear that there is barely enough data to test the computer program and certainly insufficient data to train one of the traditional text classification models as described by Pang et al. (2002). One might also question the suitability of these models when processing on units of text at the sentence level – the literature mainly discusses processing at the document level; the lexical clues necessary to perform a classification may be sparse within a single sentence. One exception is the work of Bostad (2003), though the approach seemed to suffer from processing small units of text. Additionally, it is unclear how traditional text classification models might perform when burdened with more than a few classes.

Instead it seems more promising to use Turney's PMI-IR (2002) approach. This method intuitively seems to fit well with the Two-Factor Structure of Affect (Watson and Tellegen 1985) for two reasons. The nature of PMI-IR is to quantify a piece of text's position on a one-dimensional axis. This allows for four tests to be run for each axis that needs to be quantified (that is, positive affect, negative affect, pleasantness and engagement). Secondly, this model of affect comes supplied with mood words indicating the emotions involved. These words can be used as a starting point for deriving paradigm words for each dimension that needs to be classified. The following section discusses the PMI-IR method as applied to affect recognition in more detail.

5.2. AO-PMI-IR

The purpose of PMI-IR (Pointwise-Mutual Information using Information Retrieval) is to determine the degree of synonymy between two given units of text. Indeed, PMI-IR was first introduced as a method of choosing the most probable synonym in the synonym test questions of TOEFL (Turney 2001). The algorithm was evaluated using 80 synonym test questions and obtained a score of 74%. It is rooted in the assumption that synonymous words tend to cooccur more frequently than words with less similar meaning. A large enough corpus makes calculation of pointwise-mutual information (the statistical dependence between the two words) possible with a reasonable degree of reliability.

It is this degree of synonymy that forms the basis of PMI-IR's application to the semantic orientation analysis problem (known as SO-PMI-IR), though instead of comparing a problem word with several possible choices, a problem phrase is compared with two sets of words that describe opposite poles of a paradigm (positive and negative sentiment). It is therefore a simple enough extension to compare a problem phrase with several paradigms, as defined by the Two-Factor Structure of Affect.

The following sections describe AO-PMI-IR (Affective Orientation from Pointwise-Mutual Information using Information Retrieval) – an algorithm based on PMI-IR and applied to the problem of affect recognition. It is expected that the input to this algorithm is a unit of text. In the case of this experiment, the unit of text is at the sentence level.

5.3. Candidate Phrase Selection

The first step is to identify phrases that are likely to contain affective content, using a part-of-speech tagger in conjunction with syntactic patterns. In this experiment, all the tokens have already been tagged with their part-of-speech (see Chapter 3 on Corpus Construction).

Adjectives have been identified as good indicators of subjective sentences (Wiebe et al. 2001). However, it has also been hypothesised that it may not be possible to determine the semantic orientation from a single adjective due to a limited context (Turney 2002). Turney believes that increased context reduces the need for word sense disambiguation as two-word phrases tend to be monosemous (SentimentAI 2004). Two-word phrases will also make the semantic-orientation more specific. By itself, the adjective *unpredictable* might have a negative sentiment, but placed in the domain of movie reviews, the bigram *unpredictable plot* could well have a positive sentiment (Turney 2002).

A possible way to include context, therefore, is to extract two consecutive words, where one member of the pair is an adjective or adverb and the other member provides contextual information. The patterns proposed for SO-PMI-IR are listed in Table 5.1. Singular and plural proper nouns are disregarded in order to prevent them from influencing the classification. For example, in the domain of movie reviews consider the implications of including an actor's name when determining the semantic orientation. If the actor is popular they might usually appear in positive reviews and thus be associated with positive sentiment. However, this sentiment is better associated with the actor rather than the review and is therefore not suitable grounds for influencing the semantic orientation.

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR or RBS	VB, VBD, VBN, or VBG	anything

Table 5.1: Patterns of Tags for Extracting Two-Word Phrases (Turney 2002)

For further details on the final syntactic patterns chosen for AO-PMI-IR please see the section on optimisation towards the end of this chapter.

All phrases that match the patterns are extracted and passed to the next stage to calculate their affective orientation. If no phrases can be matched then the algorithm assumes the sentence is *unclassifiable*.

5.4. Calculating Affective Orientation

PMI-IR is based on Pointwise Mutual Information (PMI) being a measure of the degree of statistical dependence between two items. PMI has been applied to several natural language processing problems including word clustering and word sense disambiguation (Manning and Schütze 2003). It is defined (Church and Hanks 1989) as:

$$\mathrm{PMI}(word_1, word_2) = \log_2\left(\frac{P(word_1 \,\&\, word_2)}{P(word_1)P(word_2)}\right)$$

PMI can be estimated using a very large corpus. The probabilities of $word_1$ and $word_2$ individually can be estimated simply by counting the number of occurrences of the word. Again the probability of $word_1$ and $word_2$ co-occurring can be estimated by counting the number of times the two words appear within a specified number of words of each other.

Turney (2002) calculates the semantic orientation of a phrase as:

$$\mathrm{SO}(phrase) = \mathrm{PMI}(phrase, positive) - \mathrm{PMI}(phrase, negative)$$

That is, the strength of the phrase's association with a set of *positive* paradigm words minus the strength of association with a set of *negative* paradigm words. (See Table 2.3 for examples of the paradigm words used). Turney performs some algebraic manipulation on the formulae above (interpreting NEAR as a cooccurrence operator and hits() as an occurrence-counting function), enabling the semantic orientation to be calculated as:

$$\mathrm{SO}(phrase) = \log_2\left(\frac{\mathrm{hits}(phrase\ \mathrm{NEAR}\ positive)\mathrm{hits}(negative)}{\mathrm{hits}(phrase\ \mathrm{NEAR}\ negative)\mathrm{hits}(positive)}\right)$$

We can determine the affect of a phrase using the same calculation four times – once for each axis of the Two-Factor Structure of Affect – and supplementing the *positive* and *negative* word sets for word sets that describe the opposing poles of each axis.

So, given a matrix **C** that holds the details of each paradigm (the class name, a vector containing words that are indicators of the high pole class and a vector of words that are indicators of the low pole of the class):

$$name = 0, high = 1, low = 2$$

$$\mathbf{C} = \begin{Bmatrix} \text{positive affect} & \text{high indicators} & \text{low indicators} \\ \text{negative affect} & \text{high indicators} & \text{low indicators} \\ \text{pleasantness} & \text{high indicators} & \text{low indicators} \\ \text{engagement} & \text{high indicators} & \text{low indicators} \end{Bmatrix}$$

And a form of SO() generalised to fit with this matrix:

$$\mathrm{O}(phrase, paradigm) = \left(\frac{\mathrm{hits}(phrase\ \mathrm{NEAR}\ paradigm_{high})\mathrm{hits}(paradigm_{low})}{\mathrm{hits}(phrase\ \mathrm{NEAR}\ paradigm_{low})\mathrm{hits}(paradigm_{high})}\right)$$

We can say that the affect class is predicted to be that which achieves the highest absolute value of O():

$$\mathrm{A}(phrase) = \underset{c_{name}}{\arg\max}_{c \in \mathbf{C}} \left|\mathrm{O}(phrase, c)\right|$$

And that the orientation of the determined class corresponds to the sign of O() – if it is greater than 0 then the orientation is *high* else the orientation is *low*.

This calculation is simply scaled up to recognise affect at the sentence level. If **S** is a vector containing all the *phrase*s of a sentence:

$$\mathrm{A}(\mathbf{S}) = \underset{c_{name}}{\arg\max}_{c \in \mathbf{C}} \left|\sum_{p \in \mathbf{S}} \mathrm{O}(p, c)\right|$$

5.5. PMI-IR and Very Large Corpora

PMI-IR was originally introduced using the World Wide Web as its source of information, as seen through the AltaVista search engine (Turney 2001; Turney 2002; Turney and Littman 2002). However, my initial tests showed that the incarnation of AltaVista in

June 2004 was unreliable. When tested with exactly the same query AltaVista would sometimes return massively different hit counts. I therefore deemed that AltaVista could not provide a suitable platform for PMI-IR at that time[1], a view held by others (Turney 2004a; SentimentAI 2004). I briefly considered Google as an alternative view of the World Wide Web, but this was discounted as the GoogleAPI returns estimated hit counts rather than actual values (Google 2004). This would add a great deal of noise into the model.

Instead, Peter Turney (2004a) proposed that the Waterloo MultiText System (WMTS) be used for the experiments. The WMTS is a distributed information retrieval system designed to handle very large corpora (WMTS 2004). It currently retrieves information using a query language called GCL from a corpus of around one terabyte in size (Turney 2004b). I developed an interface between Java and the WMTS utilising Telnet and GCL (see `WMTSClient` in Appendix D).

5.6. Paradigm Word Selection

PMI-IR depends on sensible word sets to describe the opposing extremes of each paradigm. Watson and Tellegen (1985) included lists of mood words when describing their Two-Factor Structure of Affect (see Table 2.1). However, at first glance it seemed that these would not be entirely appropriate as some words have senses that are outside of the domain of emotion (for example, the word *still* in the low negative affect paradigm predominantly has senses outside of this realm). This would bias PMI-IR as these other word senses would become significant in the test. To prevent this, the obviously ambiguous words (*active, strong, dull, content* and *still*) were dropped from consideration.

I used the remaining mood words as a foundation to select further paradigm words – they were starting points to manually traverse the WordNet synonyms (WordNet 2004). The resultant paradigm words are listed in Table 5.2.

Paradigm	High Indicators	Low Indicators
Positive Affect	ardent, aroused, avid, eager, elated, enthusiastic, excited, exhilarated, exultant, gladdened, gleeful, joyful, jubilant, peppy, prideful, rejoice, spirited, thrilled, zealous, zippy	asleep, sleepy, yawning, drowsy, inattentive, dozy, oscitant
Negative Affect	abusive, afraid, aggressive, antagonistic, antipathetic, anxious, belligerent, contemptuous, disdainful, disrespectful, distressed, disturbed, dysphoric, edgy, fearful, frightful, hostile, inimical, insulting, jittery, jumpy, nervous, nervy, offensive, opprobrious, overstrung, scornful, terrible, timorous, trepid, troubled, truculent, uneasy, unfriendly, unhappy, unquiet, upset, uptight, worried	'at rest', calm, easygoing, equable, placid, relaxed, serene, tranquil, unagitated, unruffled
Pleasantness	agreeable, amused, blessed, blissful, charitable, cheerful, chuffed, delighted, elated, elysian, euphoric, felicitous, glad, gratified, happy, kindly, paradisiacal, pleasant, pleased, satisfied, sympathetic	alone, contrite, deplorable, depressed, distressed, doleful, dysphoric, fussy, gloomy, grouchy, grumpy, infelicitous, lamentable, lonely, lonesome, melancholy, mournful, pitiful, regretful, remorseful, rueful, sad, saddening, solitary, sorrowful, sorry, unhappy
Engagement	amazed, aroused, astonished, astounded, dumbfounded, emotional, excited, flabbergasted, gobsmacked, startled, stimulated, stunned, surprised	dormant, inactive, quiescent, quiet, repose, serene, silent, tranquil

Table 5.2: Potential Paradigm Words found using WordNet

The next step was to quantitatively evaluate these potential paradigm words. This was achieved using PMI-IR once again. PMI-IR has proved its usefulness in determining the degree of synonymy between two words (Turney 2001). Consequently, I used the algorithm

[1] Cursory tests conducted at the time of writing seemed to indicate that AltaVista's problems may have been resolved.

to evaluate the similarity between each candidate word and the other words in the same set, using the calculation:

$$\text{synonymy}(word_1, word_2) = \frac{\text{hits}(word_1 \text{ NEAR } word_2)}{\text{hits}(word_2)}$$

The degree of antonymy was also measured using the similar calculation:

$$\text{antonymy}(word_1, word_2) = \frac{\text{hits}(word_2) - \text{hits}(word_1 \text{ NEAR } word_2)}{\text{hits}(word_2)}$$

Determining a final score for a given candidate word is a matter of finding the mean synonymy() score (*s*) and the antonymy() score (*a*) for the word against all other candidate members in the same word set. The score of a candidate word was calculated as:

$$\text{score}(word) = (s + a)\log \text{hits}(word)$$

The hit count of the word is used to represent that words appear more frequently are likely to provide more information (presuming of course that their word senses are limited to affect senses).

Across all sets the scores ranged from 1.59 (*oscitant*: low positive affect) to 6.36 (*happy*: high pleasantness). The mean score was 4.51 with a standard deviation of 0.91. Words that score less than 2 are cut from the experiment (the only word in this case being *oscitant*). This threshold, whilst fairly arbitrary, was chosen as it seemed reasonable given an intuitive analysis of all the scores. The mean scores for each classification are detailed in Table 5.3, showing that the word sets are reasonably consistent across classifications – the score for each class being within two standard deviations of the mean.

	Hit Count	Synonymy	Antonymy	Score
High Positive Affect	82,743	0.0018	0.9996	4.5017
Low Positive Affect	40,860	0.0205	0.9995	3.9585
High Negative Affect	107,082	0.0021	0.9987	4.3783
Low Negative Affect	68,887	0.0095	0.9990	4.2790
High Pleasantness	253,064	0.0034	0.9988	4.8188
Low Pleasantness	161,483	0.0037	0.9973	4.4259
High Engagement	161,569	0.0039	0.9987	4.5093
Low Engagement	180,239	0.0061	0.9991	4.9739
Mean	*131,991*	*0.0064*	*0.9989*	*4.5093*
Standard Deviation	*69,759*	*0.0062*	*0.0007*	*0.3262*

Table 5.3: Usefulness of Paradigm Words across Classifications

5.7. Optimisation

A subset of the FWF-Corpus (97 sentences) was set aside for use in optimising the algorithm. I used this subset to test and determine the most optimal syntactic patterns, using SO-PMI-IR to calculate the sentiment classification. SO-PMI-IR's decision was compared against that made by the expert coder from the Affect Annotation Experiment. The accuracy of each of these tests is shown in Table 5.4 - maximum values are highlighted in bold whilst minimum values are italicised.

Test	Unclassifiable	Positive	Negative	Overall
Turney's Syntactic Patterns	**90%**	*0%*	21%	**70%**
Adjectives, Adverbs and Verbs with 0.5 threshold	70%	14%	32%	59%
Adjectives or Adverbs with 0.25 threshold	73%	14%	21%	57%
Adjectives Only	66%	**43%**	*16%*	55%
Adjectives or Adverbs	49%	29%	37%	45%
Adjectives, Adverbs or Verbs with 0.25 threshold	46%	29%	37%	45%
Adjectives, Adverbs or Verbs with 0.25 threshold and stemmed	41%	29%	26%	37%
Adjectives, Adverbs or Verbs with 0.1 threshold	28%	29%	**63%**	35%
Verbs Only	24%	29%	58%	31%
Adjectives, Adverbs or Verbs	*17%*	**43%**	**63%**	*28%*

Table 5.4: Syntactic Patterns Optimisation

Turney's syntactic patterns produced the highest accuracy; however this is due to the patterns often failing to match any phrases within a test sentence. When this happens the algorithm assumes the sentence is *unclassifiable*. A goal of this experiment is to attempt to recognise classifiable sentiment and affect – it therefore seemed an acceptable trade-off to reduce the accuracy of *unclassifiable* in favour of increasing the accuracy of regular classes. The syntactic patterns chosen were single adjectives or adverbs (underlined) as they appeared to achieve similar accuracy across all three classes.

I had hoped to perform similar optimisation testing for AO-PMI-IR and the affect classification. However the distribution of affect classes within the optimisation subset was far too sparse for this to be feasible.

For this same reason it was not possible to optimise the affect paradigm words. I wanted to determine if PMI-IR could suffer from overfitting, as observed in some machine-learning models (Manning and Schütze 2003). Over-training models (in this case providing too many paradigm words) can result in reduced accuracy. This problem could have been analysed with the optimisation subset with AO-PMI-IR and iteratively increasing the number of paradigm words used. The choice of which words to add would have been made using the scores calculated for the paradigm words, as detailed in the section above on Paradigm Word Selection.

The next chapter presents the results of these algorithms as applied to the FWF Corpus sentences.

Chapter 6 Results and Discussion

This chapter presents the results of comparing the classifications made by the SO-PMI-IR and AO-PMI-IR algorithms discussed in the previous chapter with that made by the expert coder of the Affect Annotation Experiment.

6.1. Experiment Baselines

Two baselines are considered for this experiment. Baseline 1, a standard of computational linguistics when there are no previous experiments to compare against, uses prior knowledge of the distribution of classes and always assumes the class that accounts for the highest proportion of sentences. Baseline 2 simulates the accuracy that would occur by choosing a class completely at random by performing matches according to the distribution observed in the corpus (see `BaselineSimulator` in Appendix 4). Both baselines are described by tables 6.1 (for sentiment) and 6.3 (for affect).

Baseline 1 is perhaps the most useful in terms of evaluating an affect recognition algorithm for its usability as an applied technology. Baseline 2 instead reflects that it is more interesting, academically speaking, to achieve higher accuracy in the regular classes (those other than *unclassifiable*). Baseline 2 also seems more appropriate as the algorithm has been optimised for increased accuracy in the regular classes – *unclassifiable* is not as of much interest as the other classes. Given this interest, we might also consider baselines that disregard the irregular class. The reconsidered baselines are shown in tables 6.2 and 6.4, for sentiment and affect respectively.

6.2. SO-PMI-IR Results

I applied the original SO-PMI-IR algorithm to the FWF-Corpus and compared the results with the sentiment annotations made by the expert coder described in Chapter 3. This achieved an overall accuracy of 42.15%. The results are shown in table 6.1 (highest accuracies for a given class are highlighted in bold whilst the lowest are italicised in tables 6.1 through 6.4).

	SO-PMI-IR	Baseline 1	Baseline 2
Unclassifiable	39.54%	**100.00%**	*33.33%*
Positive	**54.67%**	*0.00%*	33.33%
Negative	**43.42%**	*0.00%*	33.33%
Overall	42.15%	**66.67%**	*33.33%*

Table 6.1: SO-PMI-IR Results

When compared against the reconsidered baselines with only regular classes, the accuracy becomes 47.14%, as shown in Table 6.2.

	SO-PMI-IR	Baseline 1	Baseline 2
Positive	**54.67%**	*0.00%*	50.00%
Negative	*43.42%*	**100.00%**	50.00%
Overall	*47.14%*	**67.59%**	50.00%

Table 6.2: SO-PMI-IR Results (regular classes only)

In both cases SO-PMI-IR under-performs the informed Baseline 1. However, when comparing against the likelihood of simply agreeing by chance (Baseline 2) the results indicate that the algorithm is making educated choices when considering all possible classes. Disappointingly it appears that the algorithm's classifications are fairly random when considering only regular classes. This might be attributed to the highly ambiguous nature of the test sentences; the classifications of *positive* and *negative* may be too abstract.

6.3. AO-PMI-IR Results

AO-PMI-IR, as discussed in the previous chapter, was evaluated against the FWF-Corpus by comparing the algorithm's classifications with the annotations made by the expert coder, achieving an accuracy of 32.78%. The results are shown in Table 6.3.

	AO-PMI-IR	Baseline 1	Baseline 2
Unclassifiable	37.73%	**100.00%**	*11.11%*
High Positive Affect	**16.13%**	*0.00%*	11.11%
Low Positive Affect	**45.45%**	*0.00%*	11.11%
High Negative Affect	**19.75%**	*0.00%*	11.11%
Low Negative Affect	**33.33%**	*0.00%*	11.11%
High Pleasantness	**13.33%**	*0.00%*	11.11%
Low Pleasantness	**18.18%**	*0.00%*	11.11%
High Engagement	9.09%	*0.00%*	**11.11%**
Low Engagement	*0.00%*	*0.00%*	**11.11%**
Overall	32.78%	**73.78%**	*11.11%*

Table 6.3: AO-PMI-IR Results

The AO-PMI-IR also under-performs the informed baseline, but again out-performs an algorithm of choosing classes at random.

When compared against the reconsidered baselines with only regular classes, the accuracy drops significantly to 19.44%, as shown in table 6.4, but is still above Baseline 2 of random choice. This is because the algorithm failed to identify many examples of the most populous class of High Negative Affect. However on the whole the results seem to indicate that the algorithm makes very-slightly informed decisions about affect classifications.

	AO-PMI-IR	Baseline 1	Baseline 2
High Positive Affect	**16.13%**	*0.00%*	12.50%
Low Positive Affect	**45.45%**	*0.00%*	12.50%
High Negative Affect	19.75%	**100%**	*12.50%*
Low Negative Affect	**33.33%**	*0.00%*	12.50%
High Pleasantness	**13.33%**	*0.00%*	12.50%
Low Pleasantness	**18.18%**	*0.00%*	12.50%
High Engagement	9.09%	*0.00%*	**12.50%**
Low Engagement	*0.00%*	*0.00%*	**12.50%**
Overall	19.44%	**45.73%**	*12.50%*

Table 6.4: AO-PMI-IR Results (regular classes only)

The failure of the algorithms to beat Baseline 1 can perhaps be attributed to the choice of affect model. While a reasonably strong agreement was exhibited among the annotators, it is important to remember that these were naïve annotators, without knowledge of the affect model used. This makes even the collective decisions questionable. Several annotators provided feedback, commenting that the classes presented were somewhat ambiguous and often resorted to choosing *unclassifiable* as they were unable to decide between two or more regular classes. These problems call the choice of affect model into question. Perhaps it would have been more sensible to choose one with distinct classes – ones which were obvious to the layperson – rather than one with scalar axes. Alternatively, radio buttons could have been replaced with sliding bars to set numerical values for each class, but this would still leave the classes open to a wide possibility of interpretations by the naïve annotators.

The high proportion of *unclassifiable* annotations also presented a problem to the algorithm as the syntactic patterns chosen were optimised for the extraction of the regular classes. A more sensible approach might have been to use existing subjectivity classification techniques to first determine if a sentence described some sort of affect at all. Any sentences deemed to be subjective could then be passed to the sentiment and affect classification algorithms. The current state-of-the-art techniques recognise subjectivity with an accuracy of 93.9% (Wiebe et al. 2004). Assuming this could be reproduced with the test data used in this experiment, the projected accuracy of the processes is near to 79.4% (SO-PMI-IR) and 75.43% (AO-PMI-IR), simply from pre-processing sentences for subjectivity. This projected

accuracy beats both baselines when all classes are considered, but the results for the regular classes are unaffected – further work needs to be carried out in order to refine the classification process.

One aspect that needs to be considered is the way language can modify the semantics of discourse. An obvious example is that of negation. Words such as *no*, *not*, *never*, *none*, *nowhere*, and *nothing* tend to flip the affect of the following words in the same phrase. Pang et al. (2002) pre-processed for negation by tagging all words between the negation and the following punctuation mark with NOT_. However, they found that this resulted in only minor improvements.

Other modifying words can strengthen the affect represented (e.g. *extremely* or *hardly).* Other more subtle modifiers are modal operators. Modal operators can set up a possible situation – for example, consider the sentence *'If Marie studied harder she would be a promising artist'*. The use of the word *if* implies that Marie does not study hard, while one might infer from the use of *would* that she is not currently a promising artist. The current version of AO-PMI-IR would disregard these subtle nuances of language and simply interpret the adjectives of *harder* and *promising* and ignore the implications of the modal operators. These and other modifiers of affect and sentiment have been described as *Contextual Valence Shifters* (Polanyi and Zaenen 2004), and need to be considered in any future attempt to build an accurate affect classification algorithm.

6.4. Misclassifications

Table 6.5 details the distribution of mismatched classifications for SO-PMI-IR. Rows indicate the class chosen by the expert coder, whilst columns indicate the class chosen by SO-PMI-IR when it mismatched the expert coder. These results seem to indicate that the mismatching occurring is similarly distributed throughout the three classes – all percentages are within 1.1 standard deviations of the mean.

Table 6.6 shows somewhat different circumstances for AO-PMI-IR mismatches. Here it is clear that the class of *Low Positive Affect* accounts for a remarkable number of misclassifications. This high percentage of misclassifications of *Low Positive Affect* perhaps explains why the highest accuracy (45.45%) was also achieved in this class – it appears that the algorithm is biased towards choosing this classification.

This might be attributed to the nature of the *Low Positive Affect* class. Looking at the original mood words (see Table 2.1) we can see that the *Low Positive Affect* class is itself somewhat ambiguous. Whereas the majority of the other regular classes are based on mood words that describe some emotional state, in this class the model's authors seem to be describe a lack of emotional state – that is, being asleep. It could be that inclusion of this common activity has biased AO-PMI-IR (for example falling asleep after an outpouring of emotion might be a repeated subject of discourse). The state of being asleep could be linguistically associated with presence of some emotion rather than lack of affect.

It is encouraging however, that the table also shows that the algorithm does not tend to mismatch against opposite poles of the same paradigm (for example, *High Negative Affect* sentences are rarely misclassified as *Low Negative Affect*). The cells that indicate this are highlighted in bold.

6.5. Accuracy versus Annotator Agreement

As mentioned in Chapter 4, each of the expert coder's classifications were rated according to the amount of agreement shown by the human annotators. It was hypothesised that the accuracy of PMI-IR might increase for annotations that exhibit a higher level of agreement. Figure 6.1, showing how the accuracy of the algorithm changes as the agreement scores increase, depicts a trend which appears to agree with the hypothesis.

This trend can also be interpreted as further evidence that the algorithm is making informed rather than random decisions.

		SO-PMI-IR		
		Unclassifiable	*Positive*	*Negative*
Expert Coder	*Unclassifiable*	-	57.79%	42.21%
	Positive	58.82%	-	41.18%
	Negative	39.53%	60.47%	-

Table 6.5: SO-PMI-IR Misclassifications

	AO-PMI-IR								
	Unclassifiable	*High Positive Affect*	*Low Positive Affect*	*High Negative Affect*	*Low Negative Affect*	*High Pleasantness*	*Low Pleasantness*	*High Engagement*	*Low Engagement*
Expert Coder *Unclassifiable*	-	3.31%	45.70%	16.89%	4.97%	2.98%	4.30%	15.89%	5.96%
High Positive Affect	57.69%	-	**11.54%**	3.85%	7.69%	0.00%	0.00%	15.38%	3.85%
Low Positive Affect	100.00%	**0.00%**	-	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
High Negative Affect	29.23%	4.62%	44.62%	-	**1.54%**	0.00%	4.62%	10.77%	4.62%
Low Negative Affect	25.00%	0.00%	50.00%	**0.00%**	-	0.00%	0.00%	25.00%	0.00%
High Pleasantness	15.38%	0.00%	46.15%	23.08%	7.69%	-	**0.00%**	0.00%	7.69%
Low Pleasantness	22.22%	11.11%	44.44%	5.56%	0.00%	**5.56%**	-	5.56%	5.56%
High Engagement	50.00%	0.00%	10.00%	20.00%	0.00%	0.00%	10.00%	-	**10.00%**
Low Engagement	0.00%	0.00%	66.67%	33.33%	0.00%	0.00%	0.00%	**0.00%**	-

Table 6.6: AO-PMI-IR Misclassifications

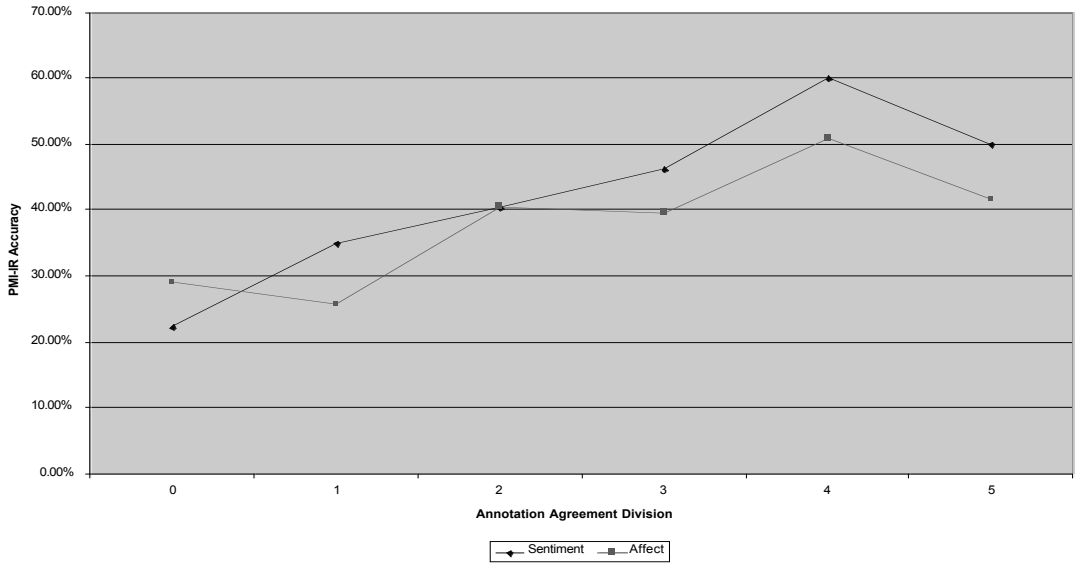

Figure 6.1: Accuracy of PMI-IR versus Annotator Agreement

6.6. Affect Lexicon Analysis

An output of AO-PMI-IR was to produce a small affect lexicon containing each adjective and adverb encountered in the FWF Corpus and its associated scores for each of the paradigms. Some examples from this lexicon are listed in Table 6.7.

	Positive Affect	*Negative Affect*	*Pleasantness*	*Engagement*
abusive	0.15	6.26	-1.04	1.77
beautifully	0.05	-2.17	0.50	-1.30
gently	-2.70	-2.07	0.10	-1.20
half-arsed	0.00	0.00	-2.53	0.00
henceforth	2.19	-0.02	1.13	-1.19
on	-0.14	0.08	0.05	0.00

Table 6.7: Examples from the Affect Lexicon

This table provides further qualitative evidence that AO-PMI-IR can inform a process classifying affect. For example, it makes sense intuitively that the adjective *abusive* be highly indicative of high negative affect and slightly indicative of unpleasantness and engagement. Other examples do not make as much sense. The word *henceforth*[*] is apparently indicative of high positive affect and slightly indicative of pleasantness and disengagement; one would expect *henceforth* to be a fairly neutral word with regards to affect.

The word *on*[*] provides another example of a word that should be neutral in terms of affect, and the AO-PMI-IR scores agree. However, the word still indicates a very small presence of semantic similarity to the affect classes. This would suggest that some degree of thresholding of scores would benefit the algorithm's accuracy.

When presenting the Two-Factor Structure of Affect (Watson and Tellegen 1985), the authors asserted that the axes of Pleasantness and Engagement could be derived from compounding the values for Positive Affect and Negative Affect. For example, Disengagement is a combination of Low Positive Affect and Low Negative Affect. The affect lexicon presents an opportunity to verify this assertion based on linguistic data. The tuple for 'gently' seems to agree with this theory, being of Low Positive Affect, Low Negative Affect and Low Engagement. However, it would be interesting to carry out thorough experiments to validate this idea statistically.

The final chapter briefly reviews the methods and the results of the experiment before proposing some possible avenues of exploration to improve the accuracy and considering some potential application areas of sentiment and affect classification technology.

[*] The words *henceforth* and *on* may not strictly be adjectives, but they have been used in the algorithm as they were identified as such by RASP. WordNet concurs with this analysis.

Chapter 7 Conclusions

This dissertation has presented a project that attempted to classify the sentiment and affect represented in sentences in text. A small corpus of 759 sentences from the domain of fiction was constructed and manually annotated with classes for sentiment and affect. An algorithm called AO-PMI-IR, based on Pointwise Mutual Information and Information Retrieval, was described and tested using the annotated corpus. These tests showed an accuracy of 32.78%. This accuracy was below a baseline informed by prior knowledge of the distribution of classifications, but above a random-choice baseline, indicating that the algorithm makes slightly-educated decisions about classifications. The algorithm can perhaps inform a larger-scale process that includes consideration of other measures related to sentiment and affect.

Future work should include the implementation of subjectivity-detection techniques in order to extract units of text that are likely to be unclassifiable (the category that accounts for the vast majority of sentences, according to the affect annotation experiment). To identify the other classes an algorithm will need to account for the way language can modify the representation of affect (for example, negation and modal operators). Similarly, sarcasm and other forms of irony present a major hurdle that must be overcome before accurate recognition of emotions can be achieved.

Another area of research that might further inform affect-recognition algorithms may be Speech Generation. In an effort to make synthesised speech sound more natural, some researchers are attempting to generate prosodic information from text (e.g. Shih and Kochanski 2001). Since listeners glean much of the affective content from the prosody of speech it would be interesting to analyse how prosody correlates with the affective content of text.

Future developments may need to consider the use of a different model of affect. The 'Two-Factor Structure of Affect' (Watson and Tellegen 1985) used in this experiment appeared to be highly ambiguous to the naïve coders making the original annotations. However, the small corpus presented in this dissertation may contain clues to any syntactic patterns that are prevalent in affective language – it should be analysed to see if these patterns exist.

It would also be interesting to experiment with similarity measures other than PMI-IR (e.g. cosine, Jensen-Shannon, α-skew, confusion probability, Jaccard's coefficient or Lin's Measure (Weeds et al. 2004), or similarity based on context (Curran 2003)). These methods may potentially be applied to find the affective orientation of words.

Technology utilising algorithms to classify emotion might benefit users of electronic communication by warning of inappropriate content in formal domains. It might also aid personal communication by verifying that emotion represented in a message is what was intended. Users of such technology could potentially avoid miscommunications of meaning and intent. On a lighter note the technology could be used to automatically generate the 'emoticons' that are widely used in online instant messaging software. Reliable recognition of emotional language could also aid the existing technologies of information extraction, summarisation and question-answering applications.

Recognition of emotion also has implications to the field of human-machine interaction, and the emerging field of Expressive Artificial Intelligence (the convergence of artificial intelligence research with artistic practice (Mateas 2001)). One particular area of Expressive AI that stands to benefit is that of interactive drama – accurate recognition of affect would provide an ability to appropriately respond to and manipulate a human participant's emotional engagement with the drama.

In conclusion, there is an encouraging scope of avenues of research into the computational recognition of affect in language and a variety of innovative applications using such affect recognition in the interaction between computer systems and humans.

Bibliography

Beineke, P., Hastie, T. and Vaithyanathan, S. 2004. The Sentimental Factor: Improving Review Classification via Human-Provided Information. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*. Barcelona, Spain. July 21-26, 2004.

Bostad, T. 2003. *Detecting Sentiment Shift for Improved Automatic Sentiment Classification.* MPhil dissertation. Supervised by Teufel, S. University of Cambridge.

Briscoe, T. and Carroll, J. 2002. Robust Accurate Statistical Annotation of General Text. *Proceedings of the 3rd International Conference on Language Resources and Evaluation.* Las Palmas, Gran Canaria.

Carletta, J. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics.* 22(2):249-254.

Carletta, J., McKelvie, D., Isard, A, Mengel, A., Klein, M., and Møller, M.B. 2002. A generic approach to software support for linguistic annotation using XML. *Readings in Corpus Linguistics*, Sampson, G. and McCarthy, D. (eds). London and New York: Continuum International.

Church, K.W. and Hanks, P. 1989. Word Association Norms, Mutual Information and Lexicography. *Proceedings of the 26th Annual Conference of the Association for Computational Linguistics.* New Brunswick, New Jersey, USA.

Curran, J. 2003. *From Distributional to Semantic Similarity*. PhD Thesis. School of Informatics, University of Edinburgh.

Ekman, P. 1992. An Argument for Basic Emotions. *Cognition and Emotion*. 6, 169-200.

Finn, A., Kushmerick, N, and Smyth B. 2002. Genre Classification and Domain Transfer for Information Filtering. *Proceedings of the European Colloquium on Information Retrieval Research.* Glasgow.

Google 2004. *Google Web APIs.* http://www.google.com/apis/. Accessed 9th June 2004.

Grefenstette, G., Qu, Y., Evans, D. A., and Shanahan, J.G. 2004. Validating the Coverage of Lexical Resources for Affect Analysis and Automatically Classifying New Words along Semantic Axes. *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. March 22-24, 2004. Stanford University.

Kobayashi, N., Unui, K., Matsumoto, Y., Tateishi, K., and Fukusmia, T. 2004. Collecting Evaluative Expressions for Opinion Extraction. *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications.* March 22-24, 2004. Stanford University.

Manning, C., and Schütze, H. 2003. *Foundations of Statistical Natural Language Processing*. MIT Press, ISBN 0-262-13360-1.

Mateas, M. 2001. Expressive Artificial Intelligence. *Leonardo: Journal of the International Society for Arts, Sciences and Technology.* 34 (2).

McCallum, A. and Nigam, K. 1998. A Comparison of Event Models of Naïve Bayes Text Classification. *AAAI/ICML-98 Workshop on Learning for Text Categorization.*

Nass, C. I., Stener, J. S. and Tabner, E. 1994. Computers are Social Actors. *Proceedings of the Conference on Human Factors in Computing Systems 1994 (CHI '94)*. Boston, Massachusetts, USA.

Ortony, A., Clore, G.L. and Collins, A. 1988. *The Cognitive Structure of Emotions.* Cambridge University Press.

Pang, B. and Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04).* Barcelona, Spain. July 21-26, 2004.

Pang, B., Lee, L. and Vaithyanathan, S. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*. July 6-7, 2002. University of Pennsylvania.

Passonneau, R. J. and Litman, D. J. 1990. Disambiguating cue phrases in text and speech. *Proceedings of the Thirteenth International Conference on Computational Linguistics (COLING-90)*. August 20-25, 1990. Helsinki, Finland.

Picard, R. W. 1997. *Affective Computing*. MIT Press, ISBN 0262661152.

Plutchik, R. 1980. A General Psychoevolutionary Theory of Emotion. In Plutchik, R. and Kellerman, H. (eds.), *Emotion: Theory, Research and Experience: Vol. 1. Theories of Emotion* (3-33). New York: Academic.

Polanyi, L. and Zaenen, A. 2004. Contextual Valence Shifters. Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications. March 22-24, 2004. Stanford University.

Riloff, E. and Wiebe, J. 2003. Learning Extraction Patterns for Subjective Expressions. *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing.* July 11-12, 2003. Sapporo, Japan.

SentimentAI. 2004. *Sentiment & Affect in Text Yahoo Group.* http://groups.yahoo.com/group/SentimentAI/. Founded 25th March 2004.

Shih, C. and Kocahanski, G. 2001. Synthesis of Prosodic Styles. *Proceedings of the 4th ISCA Workshop on Speech Synthesis.* Perthshire, Scotland

Stork, D. J. 2000. Open Data Collection for Training Intelligent Software in the Open Mind Initiative. *Proceedings of the Engineering Intelligent Systems Symposium EIS'2000*. Paisley, Scotland. June 2000.

Subasic, P. and Huettner, A. 2001. Affect Analysis of Text using Fuzzy Semantic Typing. *IEEE Transactions on Fuzzy Systems.* Special Issue, August 2001.

Turney, P. D. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the 12th European Conference on Machine Learning (ECML-2001).* Frieburg, Germany.

Turney, P. D. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, Pennsylvania, USA.

Turney, P. D. 2004a. *Email communication regarding PMI-IR, AltaVista and WMTS.* 9th June 2004.

Turney, P. D. 2004b. *Waterloo MultiText System: User's Guide.* 28th April 2004.

Turney, P. D. and Littman, M. L. 2002. *Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus*. National Research Council, Institute for Information Technology, Technical Report ERB-1094. (NRC #44929).

Watson, D. and Tellegen, A. 1985. Towards a consensual structure of mood. *Psychological Bulletin*, 98, 219-235.

WebExp. 2004. *WebExp Experimental Software.* http://www.hcrc.ed.ac.uk/web_exp/. Human Communication Research Centre, University of Edinburgh. Accessed 18th August 2004.

Weeds, J., Weir, D. and McCarthy D. 2004. Characterising Measures of Lexical Distributional Similarity. *Proceedings of the 20th International Conference of Computational Linguistics, COLING-2004.* Geneva, Switzerland. August 2004.

Wiebe, J. 2004. MPQA: Multi-Perspective Question Answering. *MITRE Corporation Website.* http://nrrc.mitre.org/NRRC/publications.htm#MPQA. Last updated March 3, 2004. Accessed August 3, 2004.

Wiebe, J., Bruce, R., Bell, M., Martin, M. and Wilson, T. 2001. A Corpus Study of Evaluative and Speculative Language. *Proceedings of the 2nd ACL SIG on Dialogue Workshop on Discourse and Dialogue.* Aalborg, Denmark.

Wiebe, J., Wilson, T., Bruce, R., Bell, M and Martin, M. 2004. Learning Subjective Language. *Computational Linguistics 30 (3).*

WMTS. 2004. The Waterloo MultiText Project. http://www.multitext.uwaterloo.ca. Accessed 10th June 2004.

WordNet. 2004. *WordNet: a Lexical Database for the English Language*. http://www.cogsci.princeton.edu/~wn/. Cognitive Science Laboratory, Princeton University. Accessed 2nd July 2004.