Emulating Mental State in Natural Language Generation Systems

Matthew Dockrey
University of British Columbia
mrd@cs.ubc.ca

Abstract

Emotional and psychological states are important factors in how people use language. Any natural language generation system which attempts to appear truly natural will need to take this into account. There are a variety of psychological studies describing these effects, but only a few which provide quantitative data. This paper describes PsychoGen, a natural language generation system based on one such set of studies. This system is shown to handle small domains well, but fundamentally unable to properly model complex psychological states such as schizophrenia.

1 Goals

Emotion and mental state are integral parts of human expression. Natural language generation (NLG) has mostly focused on how to express content without including these effects. While generic systems might not need it, artificial agents would often benefit from the ability to convey emotion not just in what they say but how they say it.

In order to explore this possibility, the goal of this project was to build a NLG system which changes its output based on emulated mental/emotional state. This was to follow the standard NLG pipeline format of document planning → microplanning → realization. (Reiter and Dale, 2000) This system, called PsychoGen, would present the user with an interface to set the mental state. These settings would then be used to influence word choice and phrase structure during the realization of a simple domain of propositions.

The main body of work was the writing of a basic microplanner that turns domain propositions into the structures required by the realization toolkit. The exact structure of these statements was to be based on a probabilistic context-free grammar (PCFG). The same PCFG was to be used at all times, but the probabilities attached to the rules would change depending on the mental state. Because only about 5% of the words in an average text convey emotional state (Pennebaker et al., 2003), lexicalization was not foreseen not play an important roll in the microplanner.

Due to the time and resource constraints, the goal was not to generate highly natural output, but to explore the inclusion of psychological data into the process.

2 Background

Other NLG systems have been developed to capture the effects of psychological models on language generation.

In (Van der Sluis and Mellish, 2008) the authors measured the effects of biasing simple descriptive phrases to make then sound more positive or negative. This was done manually on a sample of invented news stories. They showed that readers of the output correctly perceived the bias, but did not experience any induced emotion themselves.

The POLLy system described in (Gupta et al., 2007) is designed to generate collaborative task-oriented dialog as a aide to learning English as a second language. Its output can be biased as more or less polite according to the Brown and Levinson theory of politeness. It creates output in the context of a series of exchanges between people following a recipe. Because the Brown and Levinson theory includes a factor for the social distance between the actors, it was easy for the POLLy study to explore perceptions of politeness by simply changing the labels attached to the two speakers in the dialog it created.

Similarly, the PERSONAGE system (Mairesse and Walker, 2007) models the “Big Five” personality types: openness, conscientiousness, extraversion, agreeableness, and neuroticism. PERSONAGE creates restaurant recommendations using data abstracted from user reviews. The same review can be rendered in different ways depending on the personality type settings given. An extraverted review will be more informal and contain shorter sentences, for instance.

Both of these systems make use of involved content and sentence planners. While the system described in this paper is not as complex, it still follows their basic structure. Unlike them it bases its output on quantitative measurements of feature frequencies, instead of less rigorous psychological theories.

3 Data

In order to generate statements with different structures, data on their relative frequencies is needed. There has been a lot of work exploring the differences in language depending on various mental disorders (Pennebaker et al., 2003), but relatively little for pure emotion. Because the research has been done by many different groups, the features being described are not consistent and often lack the precision needed to be encoded into an NLG system. There has been some work done on automatically extracting these features (Mairesse and Walker, 2008), but that requires emotionally-tagged corpora that would be very difficult to acquire.

One exception is the Weintraub studies (Weintraub, 1981) which provide tabular data for the frequencies of many features, listed in figure 1. These were performed in the 1960s and 70s and the features reflect the psychological theories of that time.

This data was gathered from interviews across several unrelated study groups. Subjects were recorded as they talked to a researcher for a period of 10 minutes. The researcher did not prompt the subject or reply or react in any way during this time. This was an attempt to make the results as uniform and unbiased as possible, but it had the unfortunate side-effect of reducing the output for many of the groups, particularly younger children and depressives.

Feature frequencies are provided for a control group, different age ranges and educational levels, several psychological disorders (schizophrenics, delusionals, depressives, compulsives, binge-eaters and alcoholics) as well as the Watergate conspirators Richard Nixon, John Dean, H. R. Haldeman and John Ehrlichman (based on the Watergate transcripts). Of these, the data for schizophrenics, delusionals, depressives, compulsives and Richard Nixon were used in PsychoGen.

A study on the effect of anger was also performed. For this study *interrogatives*, *you*, *profanities* and *imperatives* were added to the feature list. This data is less reliable, however, because it is based entirely on two participants, both of whom were actors simulating anger. This study emphasizes the difficulty in inducing and evaluating emotional states, a concern echoed in (Van der Sluis and Mellish, 2008). The amount of language gathered was much lower, as both actors found it impossible to generate high-intensity anger for more than two or three minutes. The context of the language was very different as well. Instead of letting them speak about whatever they wanted, the actors were given a scenario (such as a lost hotel reservation) to which they might respond with anger.

non-personal references Any phrase which doesn't contain a reference to any person or persons known by the speaker, including them self.

I All occurrences of the pronoun "I".

we All occurrences of the pronoun "we".

me All occurrences of the pronoun "me".

negatives Any negatives such as "no", "never", "not", "nobody" or "nothing".

explainers A phrase that specifies a reason or other causal connection. There are usually indicated with cues such as "because", "in order that', etc.

evaluators Any judgment of relative merit. This can show opinions regarding areas such as right/wrong, good/bad, correct/incorrect or pleasant/unpleasant.

pauses The total number of seconds spent in pauses greater than 5 seconds in length.

qualifiers Any expression of uncertainty or vagueness such as "maybe", "kind of", "sort of" or "I think".

retractors Any phrase that contradicts the preceding statement. 'I don't like dogs, but I guess they're not all bad' would be an example.

expressions of feeling Any statement of the speaker's internal state such as their likes, fears or wishes.

Figure 1: Weintraub features

In addition to these contextual changes the data was also given in a different format, making integration with the other conditions very difficult. Also, the features being described are not transformational in nature and would only have been implementable as basic templates without a much more complicated dialog-based context. For these reasons the anger condition was not included in the final version of PsychoGen.

4 Implementation

All domain knowledge in PyschGen is described in the domain.xml file. This XML file consists of *objects*, *details* and *relationships*. These roughly translate into noun phrases, adjectives and verb phrases or sentences. Both hypernymy and meronymy may be represented, as well as more task-specific information like causal relations and ownership by the agent. Tags can specify that an object is a specific or named instance as well as its gender.

Figure 2 is an example of a very simple domain. Its content could be rendered as “When I was a child, my family had a mutt named Rex. He was nice. I like dogs.” Hypernymy can be seen as Rex (ID 1) is a type of mutt (ID 2) which is a type of dog (ID 3) hierarchy. Meronymy is shown in the declaration that the agent (ID 0) is part of family (ID 7). This is realized in the output by the possessive pronoun “my”. There is no restriction on which classes come first, but all references to other elements must already exist before they are used, which is why family is defined before the agent. Time can be given in either fuzzy (childhood, past) or specific (now, -5 year) terms.

```
<domain>
<object id="7" name="family" specific="false" />
<object id="3" name="dog" specific="false" />
<object id="2" name="mutt" typeof="3" specific="false" />
<object id="1" name="Rex" gender="male" typeof="2"
proper="true" specific="true" />
<object id="0" name="agent" partof="7" />

<detail id="100" object="1" desc="nice" />

<relationship id="200" subject="7" object="1" type="have"
time="childhood" />
<relationship id="201" subject="1" object="100" type="is"
time="past" subjective="true" />
<relationship id="202" subject="0" object="3" type="like"
time="now" subjective="true" />
</domain>
```

Figure 2: Example domain

The domain is loaded into the system and turned into the equivalent Java objects. These are DomainObject, DomainDetail, DomainRelationship. In addition there are DomainTime (for the temporal location) and DomainAction objects. The later represents just a verb phrase, which can be specified in the domain as a relationship lacking a subject. Internally DomainRelationships always keep their verb phrases as DomainActions, rather or not that was how it was specified in the domain XML. Each DomainElement contains references to the other elements it uses, and realization is implemented as a single recursive call to the render method on the top-level DomainRelationship.

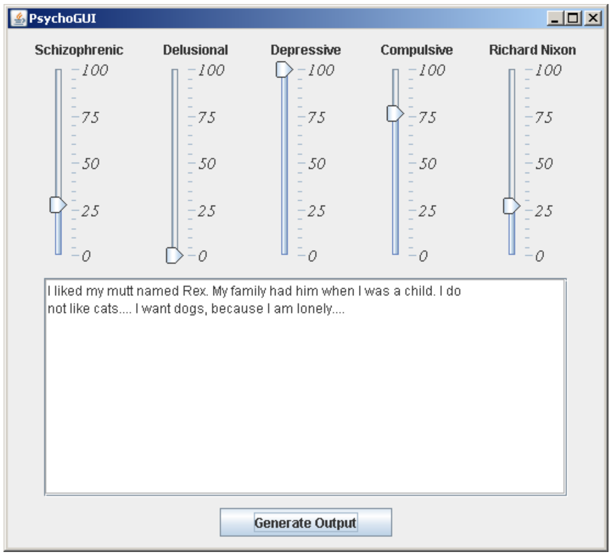
The front-end for PsychoGen presents the user with sliders for each of the psychological conditions as well as an output text window and “Generate Output” control button. (see figure 3) Before generating output, the values of the different sliders are read and converted into feature probabilities. For each sentence in the output, these probabilities are sampled, and a binary feature vector is created. This is stored in a PsychoSettings object, which is used throughout the element rendering process to determine how the domain propositions are realized.

Some features are rendered as a direct insertion of text into the current phrase. *Pauses* are simply ellipses which can be added after the sentence. *Qualifiers* are short phrases such as “I guess”, “I think”, and “kind of” which are inserted in the relevant places as sentences are rendered. This only happens on propositions which are tagged as being subjective.

Other features, such as *non-personal references*, require a more involved approach. This was only implemented for phrases of the form “I like X” and “I want X”, which are turned into “X is nice” and either “Having X would be nice” or “It would be nice to have X” respectively. Here the original elements are removed from the database and replaced with new structures entirely. This required the addition of separate verb phrases in the form of DomainActions, as well as new tags to indicate modal verbs and specify gerund and infinitive forms.

Similarly, the *negatives* feature requires the addition or subtraction of negative words, in order to turn “I don’t like X” into “I dislike X”. This is implemented as a relexicalization step using a small hard-coded set of antonyms.

The final list of implemented features was *non-personal references*, *I*, *negatives*, *explainers*, *pauses*, *we*, *qualifiers* and *expressions of feeling*.



I liked my mutt named Rex. My family had him when I was a child. I do not like cats.... I want dogs, because I am lonely....

Figure 3: Screenshot of PsychoGen

The **simplenlg** library is used for all realization. This provides a simple API for defining sentence components. It handles all morphological realization, generating the output text from the given syntactic structure. (Venour and Reiter, 2008)

Anaphora was accomplished with an internal structure for tracking noun phrase references. Whenever a DomainObject is rendered, it checks to see when the last time it was referenced, and when the last time its appropriate pronoun was used for a different object. If it had been referenced within the current or previous sentence, and the correct pronoun hasn't been used to reference another concept for the previous two sentences, then it renders as the pronoun instead of its name field. Likewise, it also checks for the first time an object is referenced, at which point it tries to add some explanatory material by adding any hypernym information available.

5 Results

PsychoGen is able to successfully represent and render moderately complicated domain propositions such as "I want a dog because I am lonely" or "It would be fun to go skiing with my neighbor next Tuesday". All of the features implemented were successful in changing the output, though sometimes only within a very limited scope such as *non-personal references*.

One major failure is the exact distribution of features. Features are applied on a sentence-by-sentence basis, resulting in certain transformations being skipped depending on the content being rendered. If the phrase being rendered contains no references to group the agent is a part of, the pronoun "we" will never be used even if the *We* feature is activated. This can only be fixed with a more complex document planner, as discussed in the Future Work section.

The following is four different realizations of a simple domain similar to the one in figure 2. The first two were rendered with the condition sliders at 0, while the second two had *schizophrenic* and *depressive* at 100.

1. I liked my mutt named Rex. When I was a kid my family had him. I guess I do not like cats. I want dogs, because I am lonely.

2. My mutt named Rex was nice. As a child my family had him. Cats are not good. Having dogs would be nice.

3. I liked my mutt named Rex. My family had him when I was a child. I do not like cats.... It would be nice to have dogs....

4. I liked my mutt named Rex.... As a child we had him.... I do not like cats.... I want dogs, because I am lonely.

Almost all of the implemented features are represented in these samples. See figure 4 for details.

non-personal references "I do not like cats" has become "Cats are not good" in #2.

I Similarly, "I want dogs" in #4 has become "Having dogs would be nice" in #2 and "It would be nice to have dogs" in #3.

we "My family" has become "we" in #4.

negatives Not represented. "I do not like cats" could have become "I dislike cats".

explainers "I want dogs *because I am lonely*" in #4. (This reason is explicitly listed in the domain.)

pauses Realized here as elipses in #3 and #4.

qualifiers "*I guess* I do not like cats" in #1.

expressions of feeling Any of the "I liked/did not like" sentences.

Figure 4: Features represented in sample output

While the system works reasonable well for shorter domains, its limitations quickly become apparent on longer ones. The lack of a proper document planner results in a series of short, boring sentences lacking connectives.

> My family had my mutt named Rex when I was a child. He was nice. Iguanas are weird. They does not make good pets. I miss having pets. I want pets, because I am lonely. My neighbor has a cat named Mr. Whiskers. He is not very friendly.... I dislike cats. I could use hobbies as well. Bird-watching would be boring. I would like to surf.

Even allowing for the odd pluralizations and lack of indefinite articles, the oddly stilted language comes across as some kind of mental pathology. This happens even when the condition sliders are all at 0, as it isn't a function of the psychological model at all. While PsychoGen works well enough for the amount of effort that went into it, it simply isn't capable of creating extended, realistic output.

6 Evaluation

Because of time and resource constraints, a proper evaluation of the results was not possible. Ideally the output would be judged by trained psychologists to see if it convincingly emulates the conditions listed. This would require more varied output at a greater quantity than this system is able to provide.

Almost as good would be to evaluate it based on existing clinical standards for the conditions being emulated. It is instructive to look at schizophrenia as an example. There has been a great deal of work done describing schizophrenia through studies of patient language. (Covington et al., 2005) What is striking is that these all tend to be very high level descriptions. For example, the Thought and Language Index items of (Liddle et al., 2002) can be seen in table 1. While no doubt useful diagnostically, they are even more ill-suited to implementation in an NLG system than the Weitraub features used here, nor are they of much help in evaluating the output.

Table 1: Thought and Language Index items

Impoverished thought/language	Poverty of speech
	Weakening of goal
Disorganized thought/language	Looseness
	Peculiar word
	Peculiar sentence
	Peculiar logic
Non-specific disregulation	Perseveration
	Distractibility

There has also been research into the automatic detection of schizophrenic language. (Strous et al., to appear) This tends to be a much lower level approach, using classical NLP techniques such as n-gram models and other easily extractable features. It is unlikely that these systems would be applicable to evaluation either, as they are looking at completely different features which wouldn't be differentiated at all in our output.

Fundamentally, a disorder like schizophrenia is largely a semantic one. Schizophenic language tends to be highly regular in phonology, morphology and syntax (Covington et al., 2005), (Kuperberg and Caplan, 2003). It isn't until meaning is considered that the pathology becomes apparent. This makes any automatic evaluation very difficult, as it requires detailed understanding to see that while "The dog bit me" is acceptable, "I bit the dog" suggests something is going wrong. The approach outlined in this paper will never be flagged as pathological under these standards, because they barely touch the semantic domain at all. The Weintraub features simply can't capture the ways in which schizophrenic (or nearly any other condition) language differs from normal language.

On a more qualitative level, the PyschoGen system generates moderately natural looking output when run on small domains. It doesn't have the complexity and variation needed to keep longer realizations interesting. It successfully implemented several of the Weintraub features, and these do noticeably change the feel of the output as the input parameters are changed. It certainly has a long way to go before it could be used in any practical context, but it proves that modeling psychological states is a NLG system is possible and not too difficult.

7 Future Work

Adding a proper document planner would be the most important next step for this system. Currently all relationships in the domain are realized, and strictly in the order they are given. There is no advanced planning to change the order, combine related concepts or drop ones that aren't needed. A document planner capable of recognizing related concepts and combining them with the appropriate conjunctions and cue words would greatly help the naturalness of the output. It is also required for a more intelligent implementation of the feature transformations. Currently they are applied on a sentence-by-sentence basis, which doesn't allow for coordination across the entire output. Because all that matters is the final distribution of the features, one could get much better, less awkward coverage by selecting potential sentences to match the features desired, instead of blindly modifying sentences one at a time.

In a similar vein, it would be a big improvement to replace the current relexicalization process with one based on WordNet instead of a hard-coded antonym list. This would also be useful in the phrase restructuring step of depersonalization, which needs to turn active verbs such as "like" and "hate" into equivalent adjectives such as "nice" and "bad".

An applet has been started to allow easy demonstrations of the system embedded in a webpage. In addition to finishing this, it would also be nice to extend the interface to allow online editing of the domain XML. Without this feature, the demonstration applet will be limited to always rendering a single domain.

The Weintraub data, while one of the few sources of data on the effect of emotional state on language, is fundamentally ill-suited to the task of informing a natural language generation system. The features it uses do not map well onto the task. Serious future work in this area will need new data sources, such as emotionally-tagged corpora which can be used to create probabilistic context free grammars. These exist, but currently only for audio and video recordings. A transcription effort for those sources could be very useful for continuing this effort.

While there probably isn't much real-world demand for an NLG system which emulated psychological disorders, one which emulates a range of emotional states could be useful in many contexts. Any intelligent agent which needs to work with a user could benefit, such as a tutoring program or an application to encourage someone to stop smoking. User profiles could help decide if a strict, aggressive approach would be more effective than a gentle, accommodating one. Potentially more lucrative would be a system for controlling video game characters. Here the system could be very dynamic, reacting to player actions appropriately. An emotional engine would be an obvious counterpart, allowing different characters to react to the same stimulus with different emotions.

8 Conclusion

This paper has described a system which models psychological states in the generation of natural language. This proved capable of realizing short domains, but the lack of a proper content planner prevents it from creating longer output that looks natural. The Weintraub features it is based upon are also shown to be lacking when it comes to modeling real speech pathologies. Better quantitative studies or tagged corpora will be needed to create a better system.

9 Appendix

Complete source code for the project will be attached to this paper. In addition, the source code and a demonstration applet[1]can be found at `http://www.cs.ubc.ca/~mrd/PsychoGen`. To be run, a copy of the simplenlg package must be included in the classpath. This is available at `http://www.csd.abdn.ac.uk/~ereiter/simplenlg/`.

References

MA Covington, C He, C Brown, L Naçi, JT McClain, BS Fjordbak, J Semple, and J Brown. 2005. Schizophrenia and the structure of language: the linguist's view. *Schizophrenia Research*, 77:58–98.

Swati Gupta, Marilyn Walker, and Daniela Romano. 2007. Generating politeness in task based interaction: An evaluation of the effect of linguistic form and culture. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 07)*.

GR Kuperberg and D Caplan, 2003. *Language dysfunction in schizophrenia*.

PF Liddle, ET Ngan, SL Caissie, CM Anderson, AT Bates, DJ Quested, R White, and R Weg. 2002. Thought and language index: an instrument for assessing thought and language in schizophrenia. *The British Journal of Psychiatry*.

François Mairesse and Marilyn Walker. 2007. PERSONAGE: Personality generation for dialogue. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague.

François Mairesse and Marilyn Walker. 2008. Trainable generation of big-five personality styles through data-driven parameter estimation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus.

JW Pennebaker, MR Mehl, and KG Niederhoffer. 2003. Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology*, 54:547–577.

Ehud Reiter and Robert. Dale. 2000. *Building natural language generation systems*. Cambridge University Press.

R Strous, M Koppel, J Fine, S Nahaliel, G Shaked, and A Zivotofsky. to appear. Automated characterization and identification of schizophrenia in writing. *Journal of Nervous and Mental Disorders*.

I. Van der Sluis and C. Mellish. 2008. Using tactical nlg to induce affective states: Empirical investigations. In *Proceedings of the Fifth International Natural Language Generation Conference (INLG 2008)*.

Chris Venour and Ehud Reiter, 2008. *A Tutorial for simplenlg*.

Walter Weintraub. 1981. *Verbal Behavior: Adaptation and Psychopathology*. Springer Publishing Company.

[1] The demonstration applet had not actually been created as of publication.