

A Pattern-Based Model for Generating Text to Express Emotion

Fazel Keshtkar and Diana Inkpen

School of Electrical Engineering and Computer Science, University of Ottawa,
Ottawa, Canada

{akeshtka,diana}@site.uOttawa.ca

Abstract. In this paper we introduce a novel pattern-based model for generating emotion sentences. Our model starts with initial patterns, then constructs extended patterns. From the extended patterns, we chose good patterns that are suitable for generating emotion sentences. We also introduce a sentence planning module, which provides rules and constraints for our model. We present some examples and results for our model. We show that the model can generate various types of emotion sentences, either from semantic representation of input, or by choosing the pattern and the desired emotion class.

Keywords: Emotion Expression, Pattern-based, Natural Language Generation, Sentiment Analysis, Surface Realization.

1 Introduction

Emotions and feelings connect us with our lives, and affect how we build and maintain the basis for interactions with people in society. These types of phenomena also take place in the virtual communities. Through the virtual environment and social networks, people can stay in touch with their relatives and friends. They exchange experiences, share opinions and feelings, and fulfill their social need for interpersonal communication, using the online world of computer-mediated communications and similar environments. Affect is an important element in social interaction. Over the past decade, issues of recognition, interpretation and representation of affect and emotion have been extensively investigated by researchers in the field of affective computing and emotion analysis. A wide range of modalities have been considered, including affect in speech, facial display, posture and physiological activity. Recently, textual information has increased, and researchers are now interested in studying different kinds of affective phenomena, including sentiment analysis, subjectivity and emotions in text.

Natural Language Generation (NLG) systems produce text or speech from a given nonlinguistic input following, an interpretation, document planning, sentence planning and surface realization tasks [10]. In this work, we focus on surface realization, the task of producing surface word strings from a nonlinguistic input specification. We introduce a pattern-based model; the user can choose a pattern, then based on the pattern variable fields, the tool will generate emotion sentences.

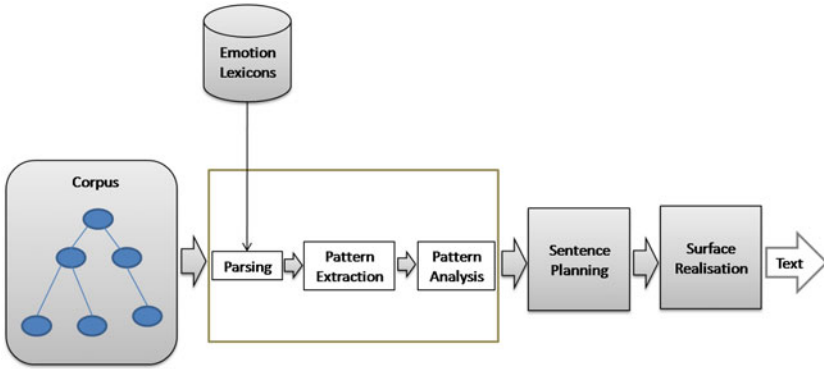


Fig. 1. Our Pattern-based Architecture for Sentence Generation

2 Our Model to Generate Emotion Sentences

The vast amount of text that is becoming available online offers new possibilities for achieving corpus-based approaches to NLG systems. We have developed a 'pattern-based model', a system that generates extraction patterns using untagged text. Our model requires only a pre-classified corpus of relevant and irrelevant texts; nothing inside the texts needs to be tagged in any way. Our goal with this model is to extract emotion patterns from the corpus, then find the best patterns to describe good semantic representations for each emotion category. For input, our model uses a semantic representation of the meaning of the input sentence to be conveyed in a different realization sentence.

During our research, for the paraphrase extraction [5], we recorded the parts-of-speech (PoS) tags, and the words surrounding the emotion seeds. We extracted pronouns and the open-class lexical terms (nouns, verbs, adjectives and adverbs) before and after each seed, for each class of emotion. They have advantages in our model: first, to find paraphrases and context similarity for paraphrase extraction; and second, to find the behavior and the formation of POS and words that surround each emotion word, which is our goal in this paper. This also helps determine the construction of emotion sentences. Figure 1 illustrates the interaction between input semantic values and sentences taken from a corpus, the pattern extraction and analysis, planning the sentence by using a finite state model, and the output text produced by the surface realization engine. Although based on a small set of examples, the combination of sentence, noun, verb, adjective and adverb patterns with the ability to change individual values, could allow the application to generate a range of sentences, broader than the target corpus.

2.1 Data Set

The starting point of our work was to provide a corpus of emotion sentences for the six emotion categories we used in our research (i.e., joy, anger, sadness, fear,

Table 1. The features that we used to extract patterns

Features	Description
F1	Sequence of PoS and Tokens of the Sentence
F2	First Verb before the Seed
F3	First Noun before the Seed
F4	Pronoun before the Seed
F5	Seed
F6	Pronoun after the Seed
F7	First Noun after the Seed
F8	First Verb after the Seed
F9	First Adjective after the Seed
F10	First Adverb after the Seed

surprise and disgust). We used the same data set that we collected for paraphrase extraction (explained in [5]). Each sentence in the target corpus might contain an emotion word, which can be used to produce a pattern of the original sentence.

2.2 Parsing and Tokenization

In our approach, rich feature representations are used to distinguish between emotions expressed toward different targets. In order to generate these representations, we did parts-of-speech tagging using the POS tagger developed by [6]. First we reviewed Table 1, and from the features in this table we used pronoun, noun, verb, adverb and adjective before and after emotion seeds, along with their tokens, for each emotion category. This step produced a syntactic analysis for each sentence. For each POS we introduced some heuristic rules to extract original patterns.

2.3 Pattern Extraction

We extracted the features shown in Table 1, to determine their frequency in the corpus. We transferred the tokenized sentences into the vector space of features. Table 2 displays statistics for four lexical classes of POS. It shows that pronouns, nouns, verbs, adverbs and adjectives are rich representation features in our data set. It also shows that, on average, 65% (out of 100% before seeds) of these features appear before emotion seeds, and 54% (out of 100% after seeds) of the time they appear after emotion seeds. In the table, L (Left) and R (Right) are parts-of-speech or tokens before emotion seeds, and parts-of-speech or tokens after emotion seeds, respectively. Previous research on word sense disambiguation in contextual analysis has identified several features that are good indicators of word properties. These include surrounding words and their POS tags, collocation and keywords in contexts [7].

Table 3 shows the percentages of these features for each emotion category, before and after emotion seeds in our data set. We examined the formation and construction of pronouns, nouns, verbs, adjectives and adverbs for each of the

Table 2. The Frequency of POS for each emotion

Emotion	Total	Pronoun	Verb	Noun	Adjective	Adverb	Total
anger(L)	7727	2061	1178	561	279	1250	5329(69%)
anger(R)	7618	1607	486	1757	190	256	2689(35%)
joy(L)	13340	2544	2275	1080	672	2487	9058(68%)
joy(R)	13542	2415	716	2961	368	877	7337(54%)
fear(L)	11988	2334	2062	952	613	2293	5961(50%)
fear(R)	12229	2177	648	2575	350	832	6582(54%)
disgust(L)	13796	2581	2444	1118	699	2570	9412(68%)
disgust(L)	14005	2441	734	3036	378	921	7510(54%)
sad(L)	16939	2858	3244	1353	905	3204	11564(68%)
sad(R)	17128	2889	881	3642	479	843	8734(51%)
surprise(L)	8406	2111	1316	602	312	1384	5725(51%)
surprise(R)	8314	1688	508	1911	222	290	4619(56%)
Total(L)	72196	14489	12519	5666	3480	13188	47049(65%)
Total(R)	72836	13217	3973	15882	1987	4019	39078(54%)

emotion categories. Based on this, we extracted the initial patterns for each emotion category.

2.4 Pattern Analysis

In this section, we analyze the patterns and explain how we can construct “good” patterns from extended patterns by extending the initial patterns. We define the notions of initial patterns, extended patterns, and good patterns.

– Initial Patterns

Based on the statistics shown in the previous sections, we were able to determine *initial pattern* for our system. We considered POS before and after emotion seeds, and we included the emotion seeds in an initial pattern. For example; 'N ES V' (N: Noun, ES: Emotion Seed, V: Verb) is an *initial pattern*. We extracted all initial patterns surrounding the emotion seeds.

– Extended Patterns

Since we intended to generate emotion sentences, and any sentence must have three main components (i.e., subject, verb, and object), it is difficult to construct emotion sentences from the initial patterns. Therefore, we extended them to create larger patterns that were suitable candidates for sentence generation. For example, from the initial pattern |V ES N|, we can construct *extended* patterns such as "N V ES N", "PR V ES N", "N V ES JJ", and many others. However, it became clear that the *extended* patterns may not be suitable candidates for sentence generation, and so we selected *good* patterns from the extended patterns.

– Good Patterns

We call p a good pattern if it can generate a grammatically-correct sentence. For clarification, we explain our method with an example. Let the initial pattern p be "V ES N". We take p and match it to the candidate sentence to

Table 3. The Percentage of POS for each Emotion

Emotion	Pronoun	Verb	Noun	Adjective	Adverb
anger(L)	27%	15%	7%	4%	16%
anger(R)	37%	11%	41%	4%	6%
joy(L)	28%	25%	12%	7%	27%
joy(R)	33%	10%	40%	5%	12%
fear(L)	39%	35%	16%	10%	38%
fear(R)	33%	10%	39%	5%	13%
disgust(L)	27%	26%	12%	7%	27%
disgust(R)	33%	10%	40%	5%	12%
sadness(L)	25%	28%	12%	8%	28%
sadness(R)	33%	10%	42%	5%	10%
surprise(L)	37%	23%	11%	5%	24%
surprise(R)	37%	11%	41%	5%	6%
Total(L)	31%	27%	12%	7%	28%
Total(R)	34%	10%	41%	5%	10%

Table 4. Some initial patterns

Initial Patterns
Noun Emotion-Seed Verb
Verb Emotion-Seed Noun
Pronoun Emotion-Seed Verb
Verb Emotion-Seed Pronoun
Adjective Emotion-Seed Verb
Adverb Emotion-Seed Verb
Verb Emotion-Seed Adjective
Verb Emotion-Seed Adverb

find the extended pattern. From the pattern p we can construct the pattern P_1 "N V ES N". An example of a correct sentence from pattern P_1 is, "Stuff looks like heaven". From p we can also construct pattern P_2 "PR V ES N", and an example of this pattern is, "It was great idea". Finally, in this example, from the initial pattern p we can construct the extended patterns P_1 and P_2 , which are *good* patterns. As shown by the examples, the *good* patterns can generate different types of emotion sentences. Figure 2 illustrates the transformation of the construction of some *initial* patterns into *extended* patterns, with examples. We followed the above method and retrieved many *good* patterns from the *extended* patterns.

2.5 Sentence Planner

Sentence planning is one the main tasks of any NLG system. It determines how the information is divided among individual sentences, and which cohesion parts (e.g., pronouns, discourse markers, verbs, etc.) should be added to

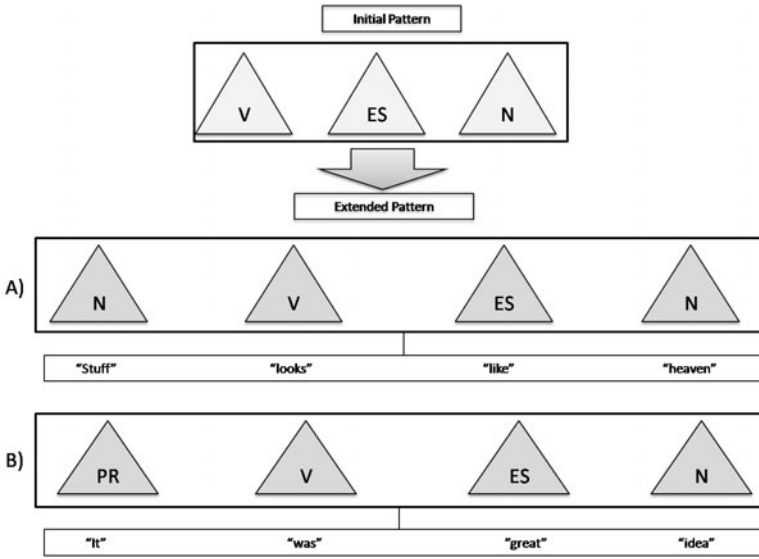


Fig. 2. Constructing Extended Pattern with Examples (ES: Emotion Seed)

Table 5. Some good patterns

Good Patterns	
N V ES N	Amy is happy girl
PR V ES N	She is nice person
PR V ES RB	they lived happily together
PR V RB ES	I am quite upset
PR V ES JJ	I am feeling smart
PR V RB ES	I am pretty good

make the text coherent and smooth [10]. In sentence planning, we define some rules for constructing patterns from the previous section to generate emotion sentences. For example, the pattern "PR V great NN" could generate the sentence "He was great idea", which is not a correct sentence. So, our rules add restrictions in order to choose a correct pronoun for subject of the sentence. We also try for agreement between the pronouns and verbs, and the coherency of the sentence. Sentence planning is a distinct phase of the generation process. It requires computing various aspects, including sentence delimitation, sentence internal organization and references.

Pattern Selection Task

- **Transforming a pattern to a sentence format.** This will manage the chosen pattern to determine the correct format for the sentence structure, in terms of subject, verb and object formation. The user can select a subject

and a verb, and the rest will be considered as object by the system. Another option is for the system to identify the subject, verb and object based on a selected pattern, and the user identify the variable parts in the system. For example, in the pattern "Pronoun Verb Emotion-Seed Noun", the Pronoun will be considered as the Subject, the Verb will be the verb and the Emotion-Seed and Noun will be considered as the Object. This will help the sentence realization module to generate an output sentence.

- **Determining correct emotion words.** This task involves selecting different syntactic alternatives for the elements of emotion seeds in the pattern, from a set of semantic representations and equivalent input texts or chosen emotion.
- **Sentence content delimitation.** If the user wants to generate more than one sentence, the sentence planner can allocate the information so that it is distributed into distinct sentences.
- **Internal sentence organization.** Within the sentence, the sentence planner must allocate the subject, specify the adjuncts, determine the order of preposition phrases, determine the subordination of relative clauses, etc.
- **Lexical choice:** This task involves selecting from a set of semantically equivalent but syntactically different alternatives.

Aggregation Task

- **Conjunction and other aggregation.** i.e., transforming (1) to (2):
 - 1) Amy is happy. Amy is nice.
 - 2) Amy is happy and nice.
- **Pronominalization and other reference.** i.e., transforming (3) to (4):
 - 3) Amy is worried. Amy has an exam.
 - 4) Amy is worried. She has an exam.
- **Introducing discourse markers.** i.e., transforming (5) to (6):
 - 5) I just saw Amy, she was sick.
 - 6) I just saw Amy, she was also sick.

The common theme behind these operations is that they do not change the information content of the text; instead they make it more fluent and easier to read. Sentence planning is important if the text needs to read fluently and, in particular, if it should appear to have been written by a human. Otherwise, there is no need to emphasize sentence planning, and the system can perform minimal aggregation.

2.6 Surface Realization

The realizer (the final NLG module) generates actual text based on decisions made by the document planner and the sentence planner (microplanner). A realizer generates individual sentences, typically from a ‘deep syntactic’ structure [9]. The realizer needs to ensure that the rules of English are followed:

- Punctuation rules: For example, the sentence: “Amy looks great, nice, and beautiful” must end with “.” not “,”
- Morphology: the plural of *box* is *boxes*, not *boxs*.
- Agreement: For example: “I am happy” instead of “I are happy”.
- Reflexive: For example: “Amy made herself happy” instead of “Amy made Amy happy”.

There are too many linguistic formalisms and rules which can be incorporated into an NLG Realizer to explain here. Some are general purpose engines such as FUF [3] and Penman [8], which can be programmed with various linguistic rules. We used SimpleNLG [4] and our Authoring Tool NLG System [1] for sentence realization, and to generate sentence. Using the pattern definitions from the previous sections, we designed a simple surface realization component for our model.

We designed a simple surface realization component for our model, using the pattern definitions from the previous section. Our surface realization module can currently accept a template as input (to be taken as a sample structure with inherited default values for the output sentence) and, optionally, parameters representing the alternative semantics of its subject, verb and object constituents. Alternatively, it is possible to specify a sentence from scratch without using an existing template as a basis [2], in a standard pattern format such as "Noun Verb Emotion-Seed Noun" or other pattern formats. We believe the latter option in the system helps to specify simpler sentence structures more conveniently, instead of having to look up an example, or find templates in the corpus. In both the template-based and pattern-based approaches the system selects a target template/pattern, then provides a set of values to fill in the template/pattern variable fields. These input values overwrite the default values provided by the template; that is, those values that were inherited from the corpus data or other lexical sources. If necessary, they are adjusted by basic agreement rules to reestablish grammaticality.

2.7 Examples and Results

Providing Variables for Fields. Depending on the type of template or pattern, our system can support five types of variables: pronouns, nouns, verbs, adjectives and adverbs. The variables for the pronoun fields can be any type of pronouns, based on the pattern or template requirement. The supported variable fields for nouns and noun phrases are: determiner type, gender, number, person, pre and post-modifiers, the noun-phrase head, proposition, or relative clause. Nouns can be chosen by the user, or from the nouns provided in Section 2.3.

For verbs or verb phrases, the variable fields are verb or verb phrase type, and can be finite, infinite, mode, verb tense or adverbial modifiers. The gender and number for verbs are not specified directly, but they can be inherited from the subject by the sentence realizer to avoid a conflicting input specification.

For adverb variable fields, our system can accept different types of adverbs, such as adverbs of manner (e.g., carefully), adverbs of time (e.g., today, next

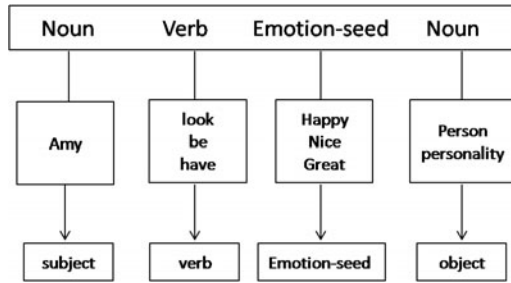


Fig. 3. An example for to generate sentence with pattern for emotion happiness

Table 6. An example of surface realization with the pattern: “Noun Verb Emotion-Seed Noun” for the emotion happiness

Amy looks happy person.
 Amy is nice person.
 Amy has nice personality.
 Stuff looks like heaven.
 It was great idea.
 They had good time.

week), adverbs of frequency (e.g., usually, occasionally), adverbs of degree (e.g., a lot, so much), and adverbs of comment (e.g., fortunately).

In terms of adjective variable fields, our system can accept different types of adjectives, such as: personal titles (e.g., Mr., Ms, Dr., etc.), possessive adjectives (e.g., my, your, his, our, their, its), demonstrative adjectives (e.g., this, that, these, those), indefinite adjectives (e.g., no, any, many, few, several), and numbers.

To generate sentences with our patterns, the user can 1) choose any pattern, 2) identify the desired emotion category (happiness, anger, fear, sadness, surprise or disgust) and 3) select the variables for pattern fields (in this case, the system can choose automatically as well). The system then 4) transform the semantic representation that is determined by the user to guide sentence formation, and generate different types of emotion sentences.

We describe how a pattern-based example works in our system below (Figure 3). The user selects a favorite pattern; here the selected pattern is “Noun Verb Emotion-Seed Noun”, and the desired emotion is “happiness”. Then the user can select “Amy” for the first Noun in the pattern, and one of the verbs “look”, “be” or “have” as the verb for the variable field Verb in the pattern. As an Emotion-Seed, suppose the user selects “happy”, “nice” and “great”. For the last field of the pattern, we consider “person” and “personality”. To generate the final output, the system will transform the combination of patterns to a sentence format which is “Subject+Verb+Emotion-Seed+Object”.

Some fields in the pattern need to be in agreement. For example, agreement between $\langle Subject, Verb \rangle$ and sometimes $\langle Subject, Object \rangle$. Also, it needs to

select proper emotion seeds to be suitable for the sentence structure, and for fluency in the final output sentence. These constraints are again performed by the sentence planner, agreement rules and the surface realization module.

With this pattern, the system can generate different types of emotion sentences. For example, we can use our Authoring Tool NLG system to generate different types of sentences with various subjects (singular, plural), verbs (with different tenses: present, past, future) and objects. By changing the emotion expression to a different emotion category (e.g., *anger*), the system is able to generate various types of emotion sentences for the *anger* emotion with the same pattern. The final surface realization for this pattern is presented in Table 6. We note here, with one pattern, the system can generate various emotion sentences for different emotion classes, but the type of emotion sentence to be generated is the user’s choice.

As any research, our work has some limitations, as well. For example, for the results in Table 6, some sentences might not be perfect English sentences. For example, “Amy looks happy girl” must be “Amy looks a happy girl”, or, for the sentence “She is nice person”, is better to have “She is a nice person“. As we can see, these sentence will be more fluent, if the determiner “a” is added. Our solutions to this are: 1) we can use a language model which can fix this deficiency, 2) we can add an extra module in our system so that the user is able to add the determiner in the generated sentence.

3 Conclusion and Future Work

This paper presented a model for sentence generation to express emotions. Our model started with initial patterns, then constructed extended patterns. From the extended patterns, we chose good patterns suitable for generating emotion sentences. We also introduced a Sentence Planning module, which provides rules and constraints for our model. Sentence Planning will also need a module that ensures that a sentence is coherent, grammatically correct and fluent.

Finally, we presented some examples and results for our model. We showed that the model can generate various types of emotion sentences, either from semantic representation of input, or by choosing the pattern and the desired emotion class. The results indicate that our system generates fluent English emotion sentences, and some that require minor modifications, though they are still usable. For future work, we plan to extend our model to cover more patterns for English sentences, and to include more emotion categories.

References

1. Caropreso, M.F., Inkpen, D., Khan, S., Keshtkar, F.: Automatic generation of narrative content for digital games. In: IEEE NLP-KE 2009, Dalian, China (2009)
2. Caropreso, M.F., Inkpen, D., Khan, S., Keshtkar, F.: Visual development process for automatic generation of digital games narrative content. In: 47th ACL-IJCNLP Workshop, WS4, Singapore, August 2-7 (2009)

3. Elhadad, M.: Using Argumentation to Control Lexicon Choice. Ph.D. thesis, Columbia University, New York, USA (1994)
4. Gatt, A., Reiter, E.: Simplenlg: A realisation engine for practical applications. In: ENLG (2009)
5. Keshtkar, F., Inkpen, D.: A corpus-based method for extracting paraphrases of emotion terms. In: 11th NAACL-HLT 2010 Association for Computational Linguistics-Human Language Technologies, WS2: Computational Approaches to Analysis and Generation of Emotion in Text, Los Angeles, CA (June 2010)
6. Klein, D., Manning, C.D.: Fast exact inference with a factored model for natural language parsing. In: Advances in Neural Information Processing Systems, vol. 15, pp. 3–10. MIT Press, Cambridge (2003)
7. Mihalcea, R.: Co-training and self-training for word sense disambiguation. In: Natural Language Learning (CoNLL 2004), Boston (May 2004)
8. Penman: The penman user guide. Tech. rep., Technical Report, Information Science Institute, Marina Del ray, CA 90292 (1989)
9. Reiter, E.: An architecture for data-to-text systems. In: Proceedings of ENLG 2007, pp. 97–104 (2007)
10. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Studies in Natural Language Processing. Cambridge University, Cambridge (2000)