

Towards Socialized Machines: Emotions and Sense of Humour in Conversational Agents

Michal Ptaszynski, Pawel Dybala, Shinsuke Higuhi, Wenhan Shi,
Rafal Rzepka and Kenji Araki
*Hokkaido University
Japan*

1. Introduction

From the beginning of computer era over half a century ago, humanity was fascinated by the idea of creating a machine substituting their mental capabilities. This New Age version of Mary Shelley's *Frankenstein* gave birth to S-F literature and was one of the motors for development of our civilisation. The mental functions digitalized as the first ones were fast processing of large numbers or sophisticated formulas for specialized fields like mathematics or physics. These functions were the most troublesome for humans, but the easiest to process mechanically. Ironically, the human mental functions said to be the most human-like, and thought of as the ones which make up a grown well-socialized man, such as a sense of humour or understanding emotions of others, were neglected in Computer Science for a long time as too subjective and therefore unscientific. With the development of the Artificial Intelligence research and the related fields, like Human-Computer Interaction (HCI) or Human Factors Design, shortly before the new millennium the door opened to the fields of research of what had been unscientific till then - Affective Computing (Picard, 1997), and Humour Processing (Binsted, 1996). When Kerstin Dautenhahn and colleagues talked about the Socially Intelligent Agents (SIA) on the AAAI Fall Symposium in 2000 (Dautenhahn et al., 2002), they signalled the need for the attempts to incorporate multiple human factors into conversational agents. However, completing the task of creating a user-friendly and human-like machine was still far ahead.

In this chapter we present some of the first practical experiments on enhancing Japanese speaking conversational agent with human factors. In our research we focused on the two important features, said to make up an intelligent and socialized man: understanding emotions of others, and a sense of humour to evoke positive attitudes in other people for better socialization (Yip & Martin, 2006). These two features are also said to be the most creative and difficult to process by machines human factors (Boden, 1998). In our research we undertake the task to incorporate these two features in a conversational agent to make it more human like. A conversational agent is enhanced with a pun generator, and a system for affect analysis. The affect analysis system uses a novel method of estimating not only the valence and type of the conveyed emotions, but also, supported with a Web-mining procedure, verifies whether the emotion is appropriate for the present context of the

conversation. The pun generator is using Web contents to generate fresh jokes each time used. We perform a number of experiments concerning the incorporation of those two features. We verify the accuracy of affect analysis system in laboratory settings, as well as in the field, during a chat of users with two conversational agents - first one using modality to enhance utterance generation based on propositions gathered from the Web, and the second one - using also automatically generated puns to better socialize with the user. We check the influence of using puns on human-computer interaction.

The outline of this chapter is as follows. First, we present the conversational agent used as a base for further experiments. Next, we describe the pun-generator, and present the ideas for its combination with the conversational agent. One of the combination methods assumes using an affect analysis system to recognize user's emotions and on its basis decide whether to generate a joke or not. We present a system for affect analysis of textual input. Finally we describe experiments with implementing those two systems - pun generator and affect analysis system - into the baseline conversational agent. The implementation of those two systems is performed first separately, and then we present the first attempt to implement both of the systems. At the end concluding remarks are presented and perspectives for further research are discussed.

2. Modalin - conversational agent as a platform for experiments

Many task-oriented conversational agents (Liu et al., 2003; Reitter et al., 2006) have been developed. Research on non-task-oriented conversational agents like casual conversation dialogue systems ("chat-bots") is on the other hand not very common, perhaps due to many amateurs who try to build naturally talking systems using sometimes very clever, but rather unscientific methods. Although there are systems with chatting abilities (Bickmore & Cassell, 2001), they concentrate on applying strategies to casual conversations rather than on their automatic generation. We believe, that the main reason is that an unrestricted domain is disproportionately difficult compared to the possible uses such a system could have. It is for example very hard to predict the contents and topics of user utterances, and therefore it is almost impossible to prepare conversational scenarios. Furthermore, scenarios need more or less specific goals to be useful. However, in our opinion, sooner or later non-task-oriented conversational agents will have to be combined with task oriented systems and used after recognizing that the user's utterance does not belong to a given task. This would lead to more natural interfaces, such as information kiosks or automatic guides placed in public places where anyone can talk to them about anything (Gustafson & Bell, 2000; Kopp et al., 2005) regardless of the role the developers intended. Well-known examples of non-task-oriented conversational agents are ELIZA (Weizenbaum, 1966) and A.L.I.C.E. Both systems and their countless imitators¹ use a lot of rules coded by hand. ELIZA is able to make a response to any input, but these responses are only information requests without providing any new information to the user. In the case of A.L.I.C.E., the knowledge resource is limited to the existing database. These examples and many other "chat-bots" need handcrafted rules, and are thus often ignored by computer scientists and rarely become a research topic. However, they have proved to be useful for e-Learning (Pietro et al., 2005) and machine

¹ Many of them have been quite successful in the Loebner Prize and the Chatterbox Challenge - competitions only for English-speaking bots, but explanations of their algorithms are not available.

learning (Araki & Kuroda, 2006) support. Therefore, building a system using automatic methods seems to be the most realistic way for unrestricted domains. Considering the large cost of developing a program capable to talk about any topic, it is reasonable to turn to the Internet - a huge and cheap source of text.

The baseline system described in this section is built upon the idea that human utterances consist of a proposition and modality (Nitta & Masuoka, 1989). The system uses an algorithm for extracting word associations from the Web and a method for adding modality to statements. The system described here works for Japanese and uses text as input and output. Though we plan to combine this project with research on voice recognition and generation, e.g., to help developing freely talking car navigation systems that by their chatting abilities could help avoiding drowsiness while driving. The general description of the system procedures in order is as follows: **1.** Extraction of keywords from user utterance; **2.** Extraction of word associations from the Web; **3.** Generation of sentence proposition using the extracted associations; **4.** Addition of modality to the sentence proposition

2.1 Extraction of keywords from user utterance

Every second millions of people update their blogs and write articles on every possible topic (Kumar et al., 2003). These are available on the Web, which can be accessed any time in a faster manner every day because of the growing efficiency of search engines. Thus, the Web is well suited to extracting word associations triggered by words from user utterance with a conversational agent. We use the Google² search engine snippets to extract word associations in real time without using pre-prepared resources, such as off-line databases. First, the system analyses user's utterances using the morphological analyser MeCab (Kudo, 2001) in order to spot query keywords for extracting word association lists. We define nouns, verbs, adjectives, and unknown words as query keywords. The reason we chose these word classes is that they, to some extent, describe the context. We define a noun as the longest set of nouns in a compound noun. For example, the compound noun *shizen gengo shori*³ (natural language processing) is treated by MeCab as three words: (*shizen* - natural), (*gengo* - language) and (*shori* - processing). Our system, however, treats it as one noun.

2.2 Extraction of word associations from the Web

The extracted keywords are used as query words in the Google search engine. The system extracts nouns from the search results and sorts them in frequency order. This process is based on the idea that words co-occurring frequently with the input words are of high relevance to them. The number of extracted snippets is 500 (value set experimentally, taking into account the processing time and output quality). The top five words of a list are treated as word associations (see Table 1). Approximately 81% of the word associations obtained using this method were judged as valid (Higuchi et al., 2008). The main reason for extracting word associations from the Web is that thanks to this method, the system can handle new information, proper names, technical terms and so on, by using only the snippets from the search engine. The word association extraction takes no more than few seconds.

² Google, <http://www.google.co.jp/>

³ All Japanese transcriptions will be written in italics.

<i>Sapporo wa samui.</i> (Sapporo city is cold.)		
Association frequency ranking:		
1	<i>yuki</i> (snow)	52
2	<i>fuyu</i> (winter)	50
3	<i>kion</i> (temperature)	16
4	<i>jiki</i> (season)	12
5	<i>Tokyo</i> (Tokyo)	12

Table 1. Examples of noun associations triggered by a user utterance.

(noun) (<i>wa</i>) (adjective)
(noun) (<i>ga</i>) (adjective)
(noun) (<i>ga</i>) (verb)
(noun) (<i>wa</i>) (verb)
(<i>so-re</i>) (<i>wa</i>) (verb)
(noun)
(adjective)
(verb)

Table 2. Proposition templates.

informative expression	frequency
<i>maa - kedo</i>	21
(Well , it can be said - but -)	
<i>maa - dana</i>	16
(Well , it can be said -)	
<i>maa - desu-ga</i>	16
(Well , it appears that -)	
<i>soko-de - desu-yo</i>	15
(Here , it is said that -)	
<i>maa - da-ga</i>	14
(Well , it can be said - but -)	
<i>maa - desu-yo</i>	12
(Well , it is that -)	

Table 3. Examples of informative expression modality

question frequency	frequency
<i>...desuka?</i>	232
(Is it that ... ?)	
<i>...kana?</i>	90
(Maybe ... ?)	
<i>...da-kke?</i>	87
(Is it right that ... ?)	
<i>...masu-ka?</i>	69
(Is it that ... ?)	
<i>...nano?</i>	68
(Is it that ... ?)	
<i>...toka?</i>	55
(... , isn't it ?)	

Table 4. Examples of question modality sentence endings

2.3 Generation of proposition using word associations

Using the associations, the system generates the proposition of a sentence reply to the user input. A proposition is an expression representing an objective statement. It is generated by applying associations to a proposition template like [(noun) (particle *wa* indicating topic) (adjective)]. We prepared 8 proposition templates manually (see Table 2). The templates were chosen subjectively after examining statistics from IRC chat logs. Our criteria for choosing the templates was that they should belong to the 20 most frequent modality patterns and to be flexible enough to fit a range of grammatical constructions, e.g., in English, "isn't it" cannot follow verbs while "I guess" can follow nouns, adjectives, and verbs. The proposition templates are applied in a predetermined order: e.g., first a template "(noun) (*wa*) (adjective)" is used; next a template "(noun) (*ga*) (adjective)" is used. However, since the generated proposition is not always a natural statement, the system uses exact matching searches of the whole phrases in a search engine to check the naturalness of each proposition. If the frequency of occurrence of the proposition is low, it is defined as unnatural and deleted. This processing is based on the idea that the phrases existing on the Web in large numbers are most probably correct grammatically and semantically. In case of discarding an unnatural proposition, the system generates another proposition in the same way. In this experiment the system used propositions for which the hit number exceeded 1,000 hits in Google. The processing proceeds as follows. The system first selects the top noun, top verb, and top adjective word associations. These are applied to the templates. If a generated proposition is judged as valid (occurrence on the Web indicates validity), it is used. If not, another template is tried until a valid proposition is found. The reason for not trying every possible combination of associations is prohibitively long processing time.

2.4 Adding Modality to the Propositions

Finally, the system adds modality to the generated proposition. By modality we mean a set of grammatical and pragmatic rules to express subjective judgments and attitudes. In our system, modality is realized through adverbs at the end of a sentence and a pair of sentence head and sentence ending auxiliary verb. This kind of modality is common in Japanese (Nitta & Masuoka, 1989).

2.4.1 Extracting Modality

There is no standard definition of what constitutes modality in Japanese. In this research we classify modality of casual conversation into questions and informative expressions. Questions are defined as expressions that request information from the user. Informative expressions are transmitting information to the user. Patterns for these modalities are extracted automatically from IRC chat logs (100,000 utterances) in advance. Modality patterns are extracted in the ways as below:

- ❖ Pairs of grammatical particles and an auxiliary verbs placed at the end of sentences are defined as ending patterns
- ❖ Sentences with question marks are defined as questions
- ❖ Adverbs, emotive words, and connectives at the beginning of sentences are defined as informative expressions
- ❖ Candidate patterns thus obtained are sorted by frequency

First, the system extracts sentence-ending patterns from IRC chat logs. If an expression contains question marks, it is classified as a question. Next, the system extracts adverbs, emotive words, and connectives from the beginning and end of sentences from the IRC logs. These pairs (beginning and end) of expressions are classified as "informative expressions". For example question expression "*desu-ka?*" (question marker) is extracted from a human utterance like "*Kyou-wa samui desu-ka?*" (Is it cold today?). An informative expression "*maa ... kedo*" is extracted from a human utterance as "*Maa sore-wa ureshii kedo*" (Well, I'm glad, but you know...). After obtaining the patterns this way, 668 for informative expressions and 396 for questions, they were filtered manually to discard the ones extracted incorrectly. The overall number of patterns obtained was 550 of the former (80%) and 292 of the latter (73%). The candidates were sorted in frequency order. The examples of modality patterns are presented in Table 3 for informative expressions and in Table 4 for questions.

2.4.2 Adding Modality

The system adds the modality from section 2.4.1 to the proposition from section 2.3 to generate the system output. This process is based on the idea that human utterance consists of proposition and modality. A modality pattern is selected randomly. For example, if the system generates the proposition "*fuyu wa samui* (winter is cold)" and selects "*iyaa ... desu-yo* (Ooh ... isn't it?)" as modality pattern, the generated output will be "*iyaa, fuyu-wa samui desu-yo* (Winter is cold, you know)". However, there is a possibility that the output is unnatural, like "*fuyu-wa samui dayo-ne* (Winter is cold, aren't it?)" , depending on the pair of proposition and modality. To solve this problem, the system uses the Google search engine to filter out unnatural output. The system performs a phrase search on the end of the sentence. If the number of search hits is higher than threshold, the output is judged as correct. If the number of hits is lower than the threshold, the output is judged as incorrect and discarded, and a new reply is generated. We experimentally set the threshold to 100 hits.

2.5 Evaluation of Modalin

We used system α , generating only the proposition, and system β , generating both proposition and modality. 5 participants used each system for 10-turn conversations and evaluated the conversations on a 5-point scale. Evaluation criteria were "will to continue the conversation" (A), "grammatical naturalness of dialogues" (B), "semantic naturalness of dialogues" (C), "vocabulary richness" (D), "knowledge richness" (E), and "human-likeness of the system" (F). Table 6 shows average scores for the evaluations of each system. System β that uses modality scored much higher than system α . In the evaluation, the participants expressed the opinion that an utterance like (xx wa yy) is unnatural and using a modality like *maa* ("well"), *moo* ("anyway") is very natural. Thus we can say that the modality expressions make the utterances of the system seem more natural. The results were considered to be very statistically significant with P value = .0032.

Evaluation criteria	System α (proposition)						System β (proposition + modality)					
	A	B	C	D	E	F	A	B	C	D	E	F
Participant a	1	3	2	2	4	2	4	4	3	4	3	5
Participant b	1	3	1	2	1	1	4	4	4	5	4	3
Participant c	1	2	1	2	1	1	1	2	1	2	1	1
Participant d	1	3	1	3	1	2	4	3	1	3	3	4
Participant e	1	4	1	1	2	1	3	2	2	4	5	4
Average	1	3	1.2	2	1.8	1.4	3.2	3	2.2	3.6	3.2	3.4

Table 6. Modalin evaluation results.

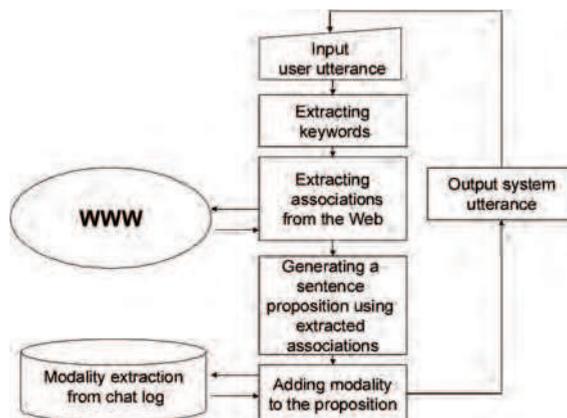


Fig. 1. Flow chart of the Modalin system.

user	<i>Nani-ka suki-na tabemono aru?</i> (What food do you like ?)
system	<i>Maa, tabemono-wa oishii-desu.</i> (Well, food tastes good.)
user	<i>Saikin-wa o-sake-mo suki-ni natte kita-n-da.</i> (Recently, I've become quite a alcohol lover)
system	<i>Demo, sake-wa yowai-no-yo-ne.</i> (But I can't drink much, oh no.)
user	<i>Yowai-n-da. Sugu yocchau-no?</i> (Oh, I see, you get loaded quickly?)
system	<i>Maa, sore-wa sugo-wa-ne.</i> (Well, it's amazing.)
user	<i>Boku-wa wari-to o-sake-wa tsuyoi hou-nan-da.</i> (In my case I guess I can drink quite a lot)
system	<i>Iya, sore-wa tsuyoi-n-de.</i> (Ooh, strong, one can say.)

Table 5. Example of a dialogue with Modalin (language errors translated close to original).

3. Pundalin – implementing humour to conversational agent

3.1 Humour as a positive factor

There are, many proofs for beneficial influence of humour on our life. For example, humour is often used as a measure to cope with negative emotions and moods, such as stress (Cann

et al., 1999), anxiety, or depression (Moran, 1996). Vilaythong et al. (2003) showed that exposing people to humorous contents, like funny videos, increased their feeling of hope and made them feel better in general. Cook & Rice (2003) provided proofs for social benefits of humour, by showing that a sense of humour in another person increases the perceived benefits of a relationship. According to Sprecher & Regan (2002), humour is also one of the main characteristics people use when choosing a partner, which means we like to interact with people with a sense of humour. Finally, Mulkey (1988) proved that we tend to use jokes when discussing difficult matters, which leads to the conclusion that humour makes conversation easier in general.

3.2 Necessity of humour in talking agents

It has been demonstrated that humans treat computers as social actors. According to SRCT (Social Response to Communication Technologies) theory, people respond to computers using the same social attitudes and behaviours they apply to humans (Reeves and Nass, 1996). This also means we expect our interaction with them go smoothly and in a natural way. Therefore, if humour enhances the interaction between humans, a similar effect should be obtained in interaction with machines.

The necessity of creating a joking conversational agent was pointed out and motivated by Nijholt (2007). However, not much has been done to actually construct such an agent. The first known attempt of this kind was made by Loehr (1996), who combined Binsted's joking system JAPE (1996) and talking agent Elmo. The results of the evaluation experiment were relatively poor, for there was barely any relevance between the user's input and the agent's humorous output. Another attempt at creating a humour-equipped agent was made by Tinholt & Nijholt (2007), who implemented a cross-reference ambiguity-based joke generator into an AIML based chat-bot. However, the opportunities for generating cross-reference jokes in daily conversation turned out to be rather rare and the impact on human involvement in the conversation could not be evaluated properly. Also, Morkes et al. (1999), checked the impact of pre-programmed (not generated) humour on a task-oriented conversation. The results showed that a humour-equipped agent was evaluated as better and easier to socialize with by human participants.

3.3 Humoroids – new class of conversational agents

Although not completely untouched (see above), the research field on humour-equipped talking agents needed to be precisely defined. The first consistent definition of such agents was proposed by Dybala et al. (2009a). His definition of this new class of agents says that humour-equipped agents, or "humoroids", are agents that are able to use humour during a conversation. He also defined two major subclasses of humoroids: task-oriented (Loehr, 1996; Morkes et al., 1999) and non-task-oriented (Tinholt & Nijholt, 2007). The agent presented here belongs to the latter type. The presence of humour is of higher importance in non-task-oriented agents, for their main purpose is to entertain human interlocutors and socialize with them during the conversation.

3.4 Punda – a pun generator for Japanese

Considering the NLP methodology, the most "computable" genre of jokes is puns. They can be found in most of the existing languages. In some, however, puns are easier to create and

thus their amount is much bigger than in others. One of such languages is Japanese, in which puns (called *dajare*) are one of main humour genres. This makes Japanese a perfect environment for pun processing research. However, although some attempts of constructing pun generating engine have been made, also in Japanese, creating a funny joking conversational system have been an unfulfilled challenge in NLP field for a long time.

PUNDA research project (Dybala et al., 2008b) is a project aiming to create a Japanese joking conversational system. As a part of this project, we developed a simple pun generating system - PUNDA Simple. This system is a simplified version of the algorithm of the main PUNDA system, which, although still under development, at its current state can be used as a pun generating support tool. Although PUNDA Simple was created for the need of this research, the main part of the algorithm is similar to the one used in the main system.

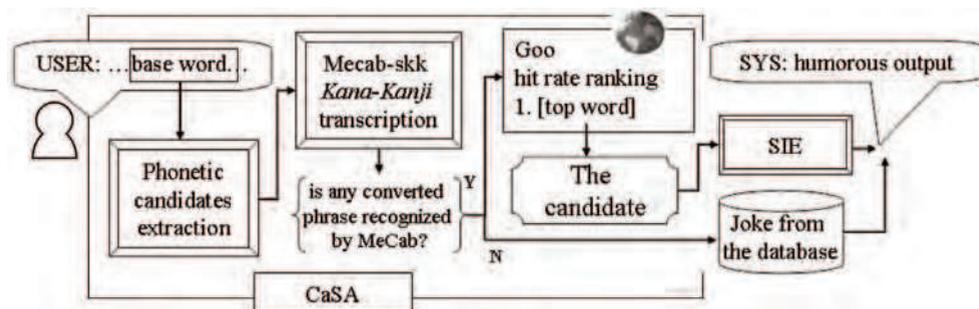


Fig. 2. Algorithm outline for PUNDA Simple joke generating engine.

3.4.1 Algorithm

The PUNDA Simple algorithm consists of two parts: Candidate Selection Algorithm (CaSA) and Sentence Integration Engine (SIE) – see Figure 2.

CaSA. In this step, the system generates a candidate for a pun. The input is a sentence, from which a base word for a pun (a word that will be transformed into a pun) is selected. The input is analysed by morphological analyser MeCab (Kudo, 2001), and if any element is recognized as an ordinary noun, it becomes the base word (a preliminary experiment proved that most of *dajare* base words are ordinary nouns). If no ordinary noun is found, one words with medium number of characters is selected randomly. Then, for the base word, pun candidates are generated using 4 generation patterns: homophony, initial mora addition, internal mora addition and final mora addition. For example, for the word *katana* (a Japanese sabre), the process goes as follows (* means one single mora):

base word: {katana}

candidates:

1. **homophony:** {katana}
2. **initial mora addition:** {*katana} (akatana, ikatana, ukatana...)
3. **final mora addition:** {katana*} (katanaa, katanai, katanau...)
4. **internal mora addition:** {ka*tana}, {kata*na} (kaatana, kaitana, kautana...)

The candidates are generated in Kana characters (one character = one mora). In the next step, for each candidate a list of possible Kanji (Japanese ideograms) transcriptions is

extracted using MeCab-skkserv Kana-Kanji Converter⁴. Then, if any of the converted character sequences was recognized by the morphological analyser as an existing word, its hit rate was checked in the Internet. The candidate with highest Goo⁵ hit rate was extracted as a pun candidate. For example, for the word *katana* the candidate would be *takatana* (a high shelf).

SIE. In this step, a sentence including the candidate extracted by CaSA is generated. To make system's response more related to the user's input, each sentence that included joke started with the pattern "[base word] to *ieba*" ("Speaking of [base word]"). The following part of the sentence is extracted from KWIC on WEB online Keyword-in-context sentences database (Yoshihira et al., 2004) - the candidate is used as a query word and the list of sentences including this word is extracted. All sentences that contain non-Japanese characters (such as Latin letters, brackets, numbers etc.) are deleted, as they may be difficult to understand without the context. Then, the shortest sentence with exclamation marks is selected (we assumed that most jokes convey some emotions, and exclamation marks are its visible sign). If no such sentence is found, the shortest sentence that ended with a period is selected. Finally, the part following the candidate is extracted and added as the second part to the system's response - see the example below:

User: - *Natsu wa atsui desu kedo, Toukyou hodo ja nai desu.* (The summer is hot, but not as hot as in Tokyo); [base word: *natsu* (summer), pun candidate: *natsukashii* (emotionally attached; loved)]

System: -*Natsu to ieba, natsukashii Nose de, kyuyuu to waiwai!* (Speaking of summer, it will be fun to meet some old friends in my beloved Nose! <a town near Osaka>)

If no appropriate sentence is found on KWIC, the response ends with "[candidate] *da yo ne*" ("it's the [candidate], isn't it") - see the example below.

User: -*Ame da kara kaimono iku no wa mendokusai ne.* (It's raining so going shopping is a bad idea) [base word: *ame* (rain), pun candidate: *ame* (a candy)]

System: -*Ame to ieba ame da yo ne.* (Rain reminds me of candies.)

When the candidates list is empty, the system selects one random pun from Sjöbergh's pun database (Sjöbergh & Araki, 2008b) - see the example below.

User: - *Sou, atarashii baito ne.* (Yeah, it's my new part-time job) [base word: *baito* (a part-time job), pun candidate: *none*]

System's response (from the pun corpus): -*Kureyon wo katte kureyon!* (Buy the pen, man!).

3.5 Pundalin – joking conversational agent

The conversational agent Modalin and the pun generator were merged to create a joking conversational system Pundalin. The merging rule was - in every third dialogue turn Modalin's output was replaced by PUNDA Simple's joke. In other words, every third user's utterance became PUNDA's input and an appropriate pun for it was generated, using the algorithm described above. This method, albeit quite simple, allowed us to check if the usage of humour improved the reception of the conversational agent in the eyes of users, and therefore system's overall performance as a companion for conversation.

⁴ MeCab-skkserv Kanji-Kana converter, <http://chasen.org/~taku/software/mecab-skkserv/>

⁵ Goo search engine, <http://www.goo.ne.jp/>

3.6 Evaluation of Pundalin

To check if humour can enhance the non-task oriented dialogue, we conducted an evaluation experiment, using Modalin as the baseline system and Pundalin as the main, humour-equipped system. As is argued by Dybala et al., (2009c), since it is users who are the “clients” of our product, in the research on dialogue systems the first person oriented evaluation is of the highest importance. It allows checking the user’s impressions of the interaction with the system in the most direct way. In the experiment, users were asked to perform a 10-turn dialogue with Modalin, and then with Pundalin. No topic restrictions were made. The utterance variety was big, however, the beginning of conversations by the users was usually very normal, like: “What did you do yesterday?”, “May I ask you a question?” or “It’s hot today, isn’t it?” There were 13 participants, 11 male and 2 female; all of them were university undergraduate students. After talking with both systems, they were asked to fill out a questionnaire about each system’s performance. The questions concerned both linguistic (B-D) and non-linguistic (A, E-H) areas of interaction: **A)** Do you want to continue the dialogue with the system?; **B)** Was the system’s output grammatically natural?; **C)** Was the system’s output semantically natural?; **D)** Was the system’s vocabulary rich?; **E)** Did you get an impression that the system possesses any knowledge?; **F)** Did you get an impression that the system was human-like?; **G)** Do you think the system tried to make the dialogue more interesting?; **H)** Did you find the conversation with the system interesting?

The replies to the questions were given on 5-point scales with explanations added. Each evaluator filled out two questionnaires, one for each system. The final, summarizing question was “Which system do you think was better?” Statistical significance of the results was calculated using the student’s t-test. The results are summarized in Table 7. The results show that the system with humour received higher scores in both linguistic and non-linguistic areas. As for the former, it may seem unusual that the presence of humour improved the system’s linguistic skills – this fact, however, could have been caused by the fact that Pundalin uses fragments of human created sentences and jokes from a data base, which naturally are more correct than those generated automatically. Also in the non-linguistic area all results point at the humour-equipped system. Users wanted to continue the conversation with Pundalin more than with Modalin; Pundalin was perceived as more human-like, knowledgeable, funny and generally better than Modalin (Dybala et al., 2008a). Results for questions A and B were found to be significant on 6% level, and for remaining questions – on 5% level. The overall compared results of Modalin and Pundalin were extremely statistically significant, with P value = .0002.

Questions	A	B	C	D	E	F	G	H	Which is better?
Modalin	2.62	2.15	1.85	2.08	2.15	2.38	1.92	2.46	15%
Pundalin	3.38	2.92	2.69	3.00	2.85	3.31	4.15	4.08	85%
Difference	0.76	0.77	0.84	0.92	0.70	0.93	2.23	1.62	
P value	> .05	> .05	< .05	< .05	< .05	< .05	< .05	< .05	

Table 7. User evaluation results for Modalin and Pundalin for detailed questions. Answers were given on a 5-point scale.

4. Implementing Emotional Intelligence in conversational agents

Developing methods for processing human emotions is one of the current issues in Artificial Intelligence. The field embracing this subject, Affective Computing, has been gathering popularity of researchers since being initiated only a little over ten years ago (Picard, 1997). The interest in this field is usually focused on recognizing the human emotions in human-computer interaction. In the popular methods, emotion recognition is focused on: facial expressions (Hager et al., 2002), voice (Kang et al., 2000) or biometric data (Teixeira et al., 2008). However, these methods, based on a behavioural approach, ignore the semantic and pragmatic context of emotions. Therefore, although they achieve good results in laboratory settings, such methods lack usability in real life. A system for recognition of emotions from facial expressions, assigning "sadness" when a user is crying would be critically mistaken, if the user was, e.g., cutting an onion in the kitchen. This leads to the need of applying contextual analysis to emotion processing. Furthermore, although it was proved that affective states should be analysed as emotion specific (Lerner & Kelter, 2000), most of the behavioural approach methods simply classify them to opposing pairs such as joy-anger, or happiness-sadness (Teixeira et al., 2008). A positive change in this tendency can be seen in text mining and information extraction approaches to emotion estimation (Tokuhisa et al., 2008; Ptaszynski et al., 2009b). However, the lack of standardization often causes inconsistencies in emotion classification. As one of the recent advances in affect analysis, it was shown that Web mining methods can improve the performance of language-based affect analysis systems (Tokuhisa et al., 2008; Ptaszynski et al., 2009b). However, in such methods, although the results of experiments appear to be positive, the two different approaches, the language-syntax based and Web mining based, are mixed. The former, comparing the information provided by the user to the existing lexicons and sets of rules, is responsible for recognizing the particular emotion expression conveyed by the user. The latter is based on gathering from the Internet large numbers of examples and deriving from them an approximated reasoning about what emotions usually associate with a certain contents. Using the Web simply as complementary mean for the language based approach, although achieving reasonable results, means not fully exploiting the great potential lying in the Web (Rzepka and Araki, 2007).

In this research we present a method capable of specifying users' emotional states in a more sophisticated way than simple valence classification. The method also contributes to standardization of the emotion classification for the Japanese language since instead of creating a new classification we apply the most reliable and coherent one available today, mentioned firstly by Ptaszynski et al. (2008) and developed further by Ptaszynski et al. (2009b), who base their classification on Nakamura's (1993) research in lexicology of emotive expressions in the Japanese language. Finally, our method does not only specify what type of emotion was expressed, but also determines whether the expressed emotion is appropriate for the context it appears in. In the method we use Ptaszynski's et al., (2009f) system for affect analysis and annotation of utterances and Shi's et al. (2008) method for gathering emotive associations from the Web. The baseline of the system presented here was first proposed by Ptaszynski et al. (2009c) and evaluated at 45% of accuracy. We improved the system in two ways. Firstly, Ptaszynski's system for affect analysis was improved with Contextual Valence Shifters to avoid confusing the valence polarity of emotive expressions. Secondly, we improved Shi's Web mining technique. The problem was it was gathering too much noise from the Internet. To solve this problem we referred to the proof provided by

Abbasi and Chen (2007), who showed that public Web services, such as forums or blogs, are rich in emotive information and thus ideal for affect analysis. Therefore we restricted the mining scope of Shi's technique from the whole Web to the contents of *Yahoo!Japan-Blogs* (blogs.yahoo.co.jp) a robust weblog service.

4.1 Definitions

Emotional Intelligence

The idea of *Emotional Intelligence* (EI) was first officially proposed by Salovey & Mayer (1990), who defined it as a part of human intelligence consisting of the ability to: I) perceive emotions; II) integrate emotions to facilitate thoughts; III) understand emotions; IV) regulate emotions. In the EI Framework (Mayer & Salovey, 1997) the first step consists of the abilities to a) identify emotions and b) discriminate between appropriate and inappropriate expressions of emotion. Salovey and Mayer (1990) argue that recognizing emotions is only the first step to acquire full scope of Emotional Intelligence and does not tell us anything about whether it is appropriate for a given situation or what reactions should be undertaken. According to Solomon (1993), the valence of emotions is determined by the context they are expressed in. For example, anger can be positive, when warranted (e.g. a reaction to a direct and deliberate offence) and negative, when unwarranted (scolding one's own children unjustly) and the reactions should be different for the two different contexts of anger. The attempts to implement the EI Framework usually do not go beyond theory (Andre et al., 2004), and the few practical attempts eventually still do not surmount the first step of recognition (Picard et al., 2001). The research presented here is an attempt to go beyond this simple approach. Following emotion recognition, their appropriateness is verified against their contexts. By providing an agent means to determine the appropriateness of emotions, we make a step towards the full implementation of EI framework in machines.

Definition and classification of emotions

Our working definition of emotions is based on Nakamura's (1993), who defines them as every temporary state of mind, feeling, or affective state evoked by experiencing different sensations. This definition is complemented by Solomon's (1993), who argues that people are not passive participants in their emotions, but rather the emotions are strategies by which people engage with the world. Since we operate on language, the above is further complemented by Beijer's (2002) definition of emotive utterances, which he describes as every utterance in which the speaker is emotionally involved, and this involvement, expressed linguistically, is informative for the listener. Nakamura (1993), proposed also a 10 type emotion classification, the most appropriate for the Japanese language: *ki* / *yorokobi* (joy, delight), *do* / *ikari* (anger), *ai* / *aware* (gloom, sorrow, sadness), *fu* / *kowagari* (fear), *chi* / *haji* (shame, shyness, bashfulness), *kou* / *suki* (liking, fondness), *en* / *iya* (dislike, detestation), *kou* / *takaburi* (excitement), *an* / *yasuragi* (relief) and *kyou* / *odoroki* (surprise, amazement).

Contextual Valence Shifters

The idea of Contextual Valence Shifters (CVS) as an application in Sentiment Analysis was first proposed by Polanyi & Zaenen (2004). They distinguish two kinds of CVS: negations and intensifiers. The group of negations contains words like "not", "never", and "not quite", which change the valence polarity of semantic orientation of an evaluative word they are attached to. The group of intensifiers contains words like "very", "very much", and "deeply", which intensify the semantic orientation of an evaluative word. So far the idea of CVS

analysis was successfully applied to Sentiment Analysis of English texts (Kennedy & Inkpen, 2005). Successful attempts on Japanese ground (Miyoshi & Nakagami, 2007) show that it is also applicable for the Japanese language. Examples of CVS negations in Japanese are grammatical structures like: *amari -nai* (not quite-), *-to wa ienai* (cannot say it is-), or *mattaku -nai* (not at all-). Intensifiers are represented by: *totemo-* (very much-), *sugoku-* (-a lot), or *kiwamete-* (extremely). The idea of CVS is applied in line with Ptaszynski's et al. (2009a) research on improving affect analysis by valence shifting. The Contextual Valence Shifting Procedure (details below) is supported further with Russell's (1980) 2-dimensional model of affect.

Two-dimensional model of affect

The idea of a two-dimensional model of affect was first proposed by Schlosberg (1952) and developed further by Russell (1980). Its main assumption is that all emotions can be described in a space of two-dimensions: valence polarity (positive/negative) and activation (activated/deactivated). An example of positive-activated emotion is excitement; positive-deactivated emotion is, e.g., relief; negative-activated and deactivated emotions are anger and gloom respectively. This way 4 emotion areas are distinguished: activated-positive, activated-negative, deactivated-positive and deactivated-negative. Nakamura's emotion types were mapped on this model and their affiliation to one of the spaces was determined. Those emotions for which the affiliation was not obvious (e.g. surprise can be both positive as well as negative, etc.) were mapped on all of the areas they could belong to. However, no emotion type was mapped on more than two adjacent fields. This grouping is then used in our system for two reasons. Firstly, in the CVS analysis procedure to specify which emotion corresponds to the one negated by a CVS phrase. Secondly, in emotion appropriateness verification procedure, for estimating whether the emotion types belong to the same area, even if not perfectly matching with the emotive associations gathered from the Web.

Example of a sentence (English translation)	Emotemes	Emotive expressions
(1) <i>Kyo wa <u>nante</u> kimochi ii hi <u>nanda!</u></i> (Today is such a nice day!)	yes	yes
(2) <i><u>Iyaa</u>, sore wa <u>sugoi</u> desu ne!</i> (Woa, that's great!)	yes	no
(3) <i>Ryoushin wa minna jibun no kodomo wo aishiteiru.</i> (All parents love their children.)	no	yes
(4) <i>Kore wa hon desu.</i> (This is a book.)	no	no

Table 8. Examples of sentences containing emotemes (underlined) and/or emotive expressions (bold type font).

4.2 Linguistic approach to emotions – the emotive function of language

The semantic and pragmatic diversity of emotions is best conveyed in language (Solomon, 1993). Therefore we designed our method to be language-based. There are different linguistic means used to inform other interlocutors of emotional states. The elements of speech used to convey emotive meaning are described by the emotive function of language (Jakobson, 1960). In Japanese it is realized lexically through such parts of speech as exclamations (Beijer, 2002), hypocoristics (endearments), vulgar language (Crystal, 1989;

Potts & Kawahara, 2004) and mimetic expressions (in Japanese: *gitaigo*) (Baba, 2003). A key role in expressing emotions is also played by the lexicon of words describing emotional states (Nakamura, 1993). The para-linguistic elements, like intonation, are represented lexically by exclamation marks or ellipsis. Ptaszynski (2006) classified the realizations of emotive function in Japanese in two general types. The first one, emotive elements (or **emotemes**), indicate that emotions have been conveyed, but not detailing their specificity. This group is linguistically realized by interjections, exclamations, mimetic expressions, or vulgarities. The second type, **emotive expressions**, are parts of speech like nouns, verbs, adjectives or metaphors describing affective states. Examples of sentences containing emotemes and/or emotive expressions are shown in Table 8. Examples (1) and (2) are emotive sentences. (1) is an exclamative sentence, which is determined by the use of exclamative constructions *nante* (how/such a) and *nanda!* (exclamative sentence ending), and contains an emotive expression *kimochi ii* (to feel good). (2) is also an exclamative. It is easily recognizable by the use of an interjection *iyaa*, an adjective in the function of interjection *sugoi* (great), and by the emphatic particle *-ne*. However, it does not contain any emotive expressions and therefore it is ambiguous whether the emotions conveyed by the speaker are positive or negative. The examples (3) and (4) show non-emotive sentences. (3), although containing an emotive verb *aishiteiru* (to love), is a generic statement and, if not put in a specific context, does not convey any emotions. Finally, (4) is a simple declarative sentence without any emotive value.

4.2.1 Defining emotive linguistic features

We defined emotemes and emotive expressions according to Ptaszynski's two-part classification. The feature set was defined in a way similar to the one proposed by Alm et al. (2005), by using multiple features to handle emotive sentences. Alm however, designed their research for English children's stories, whereas we focus on utterances in Japanese, and therefore used Ptaszynski's classification as more appropriate for our research.

Emotemes

Into the group of emotive elements, formally visualisable as textual representations of speech, Ptaszynski (2006) includes the following lexical and syntactical structures.

Exclamative utterance. The research on exclamatives in Japanese (Ono, 2002; Sasai, 2006) provides a wide scope of topics useful as features in our system. Some of the exclamative structures are: *nan(te/to/ka)-*, *-darou*, or *-da(yo/ne)*, partially corresponding to *wh*-exclamatives in English (see the first sentence in Table 8).

Interjections are typical emotemes. Some of the most representative Japanese interjections are *waa*, *yare-yare* or *iyaa* (see the second sentence in Table 8).

Casual Speech. Casual speech is not an emotem per se, however, many structures of casual speech are used when expressing emotions. Examples of casual language use are modifications of adjective and verb endings *-ai* to *-ee*, like in the example: *Ha ga itee!* (My tooth hurts!), or abbreviations of forms *-noda* into *-nda*, like in the example: *Nani yattenda yo!?* (What the hell are you doing!?).

Gitaigo. Baba (2003) distinguishes *gitaigo* (mimetic expressions) as emotemes specific for the Japanese language. Not all *gitaigo* are emotive, but rather they can be classified into emotive mimetics (describing one's emotions), and sensation/state mimetics (describing manner and appearance). Examples of emotive *gitaigo* are: *iraira* (be irritated), like in the sentence:

Omoidasenkute iraira shita yo. (I was so irritated, 'cause I couldn't remember.), or *hiyahiya* (be in fear, nervous), like in the sentence: *Juugeki demo sareru n janai ka to omotte, hiyahiya shita ze.* (I thought he was gonna shoot me - I was petrified.)

Emotive marks. This group contains punctuation marks used as textual representations of emotive intonation features. The most obvious example is exclamation mark „!“ (see Table 8). In Japanese, marks like “...” (ellipsis), or prolongation marks, like “—” or “~” are also used to inform interlocutors that emotions have been conveyed

Hypocoristics (endearments) in Japanese express emotions and attitudes towards an object by the use of diminutive forms of a name or status of the object (*Hanako* [girl's name] vs *Hanako-chan* [/endearment/]; *o-nee-san* [older sister] vs *o-nee-chan* [sis /endearment/], *inu* [a dog] vs *wanko* [doggy /endearment/]). Sentence example: *Saikin Oo-chan to Mit-chan ga boku-ura to karamu youni nattedekita!!* (Oo-chan and Mit-chan has been palling around with us lately!!)

Vulgarisms. The use of vulgarisms usually accompanies expressing emotions. However, despite a general belief that vulgarisms express only negative meaning, Ptaszynski (2006) notices that they can be also used as expressions of strong positive feelings, and Sjöbergh (2006) showed, that they can also be funny, when used in jokes, like in the example: *Mono wa mono dakedo, fuete komarimasu mono wa nanda-? Bakamono.* (A thing (*mono*) is a thing, but what kind of thing is bothersome if they increase? Idiots (*bakamono*).)

Emotive expressions

A lexicon of expressions describing emotional states contains words, phrases or idioms. Such a lexicon can be used to express emotions, like in the first example in Table 8, however, it can also be used to formulate, not emphasized emotively, generic or declarative statements (third example in Table 8). Some examples are:

adjectives: *sabishii* (sad), *ureshii* (happy);

nouns: *aijou* (love), *kyofu* (fear);

verbs: *yorokobu* (to feel happy), *aisuru* (to love);

fixed phrases/idioms: *mushizu ga hashiru* (give one the creeps [of hate]), *kokoro ga odoru* (one's heart is dancing [of joy]);

proverbs: *dohatsuten wo tsuku* (be in a towering rage), *ashi wo fumu tokoro wo shirazu* (be with one's heart up the sky [of happiness]);

metaphors/similes: *itai hodo kanashii* (pain of sadness), *aijou wa eien no honoo da* (love is an eternal flame);

4.3 ML-Ask

Based on the linguistic approach towards emotions as well as the classification of emotions, Ptaszynski et al. (2009f) constructed ML-Ask (eMotive eLements-SeeK & Analyse) system for automatic annotation of utterances with emotive information. The emotem database was gathered manually from other research and grouped into five types (code, reference research and number of gathered items in square, round and curly brackets, respectively):

1. [EX] Interjections and structures of exclamative and emotive-casual utterances (Nakamura, 1993; Oshima-Takane et al., 1995-1998; Tsuchiya, 1999; Ono, 2002). {477}
2. [GI] *Gitaigo* (Nakamura, 1993; Oshima-Takane et al., 1995-1998; Baba, 2003). {213}
3. [HY] Hypocoristics (Kamei et al., 1996). {8}
4. [VU] Vulgarisms (Sjöbergh, 2008a). {200}

5. [EM] Emotive marks (Kamei et al., 1996). {9}

These databases were used as a core for ML-Ask. We also added Nakamura's (1993) dictionary as a database of emotive expressions (code: [EMO-X], 2100 items in total). The breakdown with number of items per emotion type was as follows: *yorokobi* {224}, *ikari* {199}, *aware* {232}, *kowagari* {147}, *haji* {65}, *suki* {197}, *iya* {532}, *takaburi* {269}, *yasuragi* {106}, *odoroki* {129}.

4.3.1 Emotems analysis procedure

Based on the databases described above, a textual input utterance is analysed and emotive information is annotated. The system first determines whether an utterance is emotive (appearance of at least one emotive feature), extracts all features from the sentence, and analyses the structure of the emotive utterance. This is the system's main procedure. Examples of analysis are shown below (from top line: example in Japanese, emotive information annotation, English translation; emotems-underlined, emotive expressions bold type font, n-noun, ptl-particle, AUX-auxiliary verb, the system flow is shown on Figure 3).

- (1) *Kyo wa nante kimochi ii hi nanda !*
 Today ptl:THEM EX:nante **EMO-X:joy** day:SUBJ EX:nanda EM:!
- (2) *Iyaa, sore wa sugoi desu ne !*
EX:iyaa that ptl:THEM EX:sugoi AUX EX:ne EM:!
- (5) *Akirame cha ikenai yo !*
EMO-X:dislike EX:cha | CVS:cha-ikenai{→joy} EX:yo EM:!
- Translation: Don't cha give up!

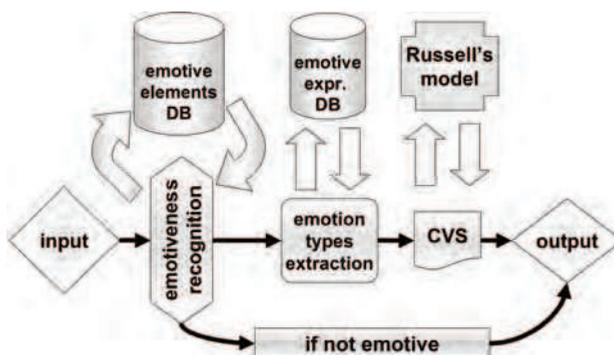


Fig. 3. ML-Ask system flow chart

4.3.2 Emotive expressions analysis procedure

In all utterances determined as emotive, the system searches for emotive expressions from the databases (Nakamura's dictionary). This procedure is used to verify:

- 1) How many of all determined emotive utterances contain emotive expressions;
- 2) If the system is capable of determining specific types of emotions in human-computer interaction. However, keyword-based extraction allowed mismatching the specific emotion types. To avoid this we applied Contextual Valence Shifters.

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

