DOI 10.3233/AIS-2010-0053
IOS Press

Multiagent system for joke generation: Humor and emotions combined in human-agent conversation

Pawel Dybala[*], Michal Ptaszynski, Jacek Maciejewski, Mizuki Takahashi, Rafal Rzepka and Kenji Araki
Graduate School of Information Science and Technology, Hokkaido University, Kita 14 Nishi 9, Kita-ku, 060-0914 Sapporo, Japan

Abstract. In this paper we present an innovative work on a multiagent joking conversational system. In our research so far we have shown that implementing humor into a chatterbot can visibly improve its performance. The results presented in this paper are the outcome of the next step of our work. They show that a multiagent system, combining a conversational agent, a pun generator and an emotiveness analysis engine, works reasonably well in interactions with users. In the setup used in this research, the emotiveness analysis agent analyses users' utterances and decides whether it is appropriate to tell a pun. Depending on the results of this analysis, the agent chooses either the pun generator, if the decision is that a joke should be told, or the non-humor-equipped agent when the decision is different. Two evaluation experiments were conducted: user (first person) focused and automatic (emotiveness-analysis-based). In both, we compared the performance of the multiagent joking system and a baseline (non-humorous) conversation agent. The results show that in both cases the humor-equipped engine was evaluated as better than the baseline agent. The results are discussed and some ideas for the future are given.

Keywords: Humor processing, emotiveness analysis, chatterbots, conversational agents, human-computer interaction

1. Introduction

In recent years, the world of computer science has finally started to appreciate the role of such ambiguous and difficult to define features as humor or emotions. Especially in the case of the latter, we can talk about an "emotiveness boom" in the field of AI. There are numerous research programs, financed by powerful institutions including the EU, aiming to create machines that would be able to "experience" (detect) or express emotions. A good example of such a venture is the LIREC Project [16], in which European researchers are working on creating artificial companions able to build long-term relationships with humans. Among many others, some areas of the project focus on emotional robots [15] or affective cues in human-robot interaction [4]. The role of emotion recognition in natural dialogue between humor and computers was emphasized by many scientists, e.g. Minker et al. [20]. Research on computers using and understanding humor may not yet be highly popular, but continues to gather more and more scientific attention (see Section 1.3).

There are numerous works in non-computer science-related areas such as psychology or sociology, describing how humorous stimuli can change or reduce particular emotions, such as stress, anxiety or boredom (current advances in this field are briefly reviewed in Section 1.2). The overall message coming from these works is that humor can make us feel better. Thus, more surprising is the fact that – to our knowledge – no existing research actually unites the two fields of computational humor and emotion recognition.

In our research we are trying to bridge this gap. Our main goal is to construct a non-task oriented

[*] Corresponding author. E-mail: paweldybala@media.eng.hokudai.ac.jp.

conversational agent (chatterbot) that would use humor in a deliberate way, i.e. would be able to react with humor to users' emotional states. Based on related works (briefly listed in Section 1.2) and on our findings so far, we expect that such a system would make the users feel better, and convert their negative emotions into positive (or at least reduce the degree of negativity). We also expect that such a system would be recognized by users as more friendly, human-like and generally better than a similar system without humor, and that conversations with our system would be more interesting.

In this paper we describe results of our recent work, in which we constructed a joke (pun) generating agent, implemented it into a conversational agent (chatterbot), and then combined it with an emotiveness detecting agent, which decides whether the humor-equipped agent should tell a joke (see Section 3 for the algorithm outlines). The emotiveness detecting agent was also used in one of the evaluation experiments (see Section 4.2), in which the records of user-agent conversations were analyzed to check users' emotive states and mood changes.

To summarize, the emotiveness detecting agent interacts with the humor-equipped agent in two ways:

- it decides if it is appropriate to tell a joke (timing detection) and
- it evaluates its performance (automatic chat logs analysis).

The results of experiments conducted to verify if our approach is correct (described in Section 5) are reasonably satisfying, and in our opinion show that we are proceeding in the right direction. The originality of this contribution is described in detail in Section 2.2.

1.1. Possible applications

One may question the importance of research on joke-generators as a whole. Ultimately, listening to jokes is fun, but how often would an average user want to do that?

In our research, we also face these questions. In fact, this is why we are so concerned with implementing humor generators into chatterbots or systems that interact with users. Including jokes in a conversation can make it better, more interesting, natural and easier to conduct (see Section 1.2), which we also showed in our previous research [7].

It may seem that creating such humor-equipped talking agents would only have one possible application, namely pure entertainment. However, this claim is not necessarily true, as machine humor is actually currently attracting the interest of, for example, car manufacturing companies working on intelligent car navigators able to entertain drivers by talking andpossibly by joking (as mentioned in Section 1.2, humorous stimuli activate the human brain). In fact, our research presented in this paper was also partially financed by one such company.

Humor-equipped conversational agents are also a subject of interest to robotics companies working on artificial companions for elderly and lonely people. As we know, humor can make us feel better, which makes it a very desirable feature for such devices.

1.2. Humor and emotions

The beneficial influence of humor on our emotions has been proved in numerous scientific works. Szabo, for instance, showed that watching a humorous video can significantly increase positive mood while decreasing emotional distress and anxiety [33]. Similarly, Vilaythong et al. [37] presented results showing that humorous stimuli (also a video) can increase human feelings of hope, while Dienstbier proved that the presence of humor can change one's perception of a boring task into an interesting one [5]. Humor was also showed to be successful in fighting depression [23] and various mood disturbances [14]. Also, a negative correlation was found between self-reported humor scales and emotional burnout [9].

The above paragraph includes only a brief summary of works showing that humor can turn negative feelings into positive. Thus, it seems natural that these beneficial features can and should be used also in such field as AI, computational intelligence, agent technologies or HCI.

1.3. Enhancing HCI with humor

As mentioned above, recent computer science has finally started to appreciate the potential of humor. Although we are still far from a "humor boom", there are some works aiming to create humor (joke) generators and enhancing HCI with funniness.

Most of the projects so far (e.g. [2,19,36]) are more or less strongly related to NLP (Natural Language Processing), as the linguistic aspect of humor is relatively easier to compute than others. In fact, there is even a well-defined genre of jokes, called "puns" or "word plays" (also "linguistic humor"), which is based on ambiguous features of language such as homophony or polysemy. A good example of a pun is the well-known "deer joke":

– What do you call a deer with no eyes?
– No-eye deer,

in which the funniness comes from the phonetic similarity between two phrases: "no idea" and "no-eye deer".

As they are based on features of language itself, puns are especially popular in the field of computational humor. Most research so far has focused on this genre and attempted to construct pun-generating engines.

Probably the best known of all works on joke generating engines is the punning riddles generator JAPE created by Binsted [2]. Using a set of symbolic rules and a large natural language lexicon, JAPE was able to produce question-answer puns, such as:

– What do you call a murderer with fiber?
– A cereal killer.

Some of the generated jokes were evaluated by schoolchildren as being similarly funny as punning riddles generated by humans.

The results of JAPE's evaluation experiment were quite impressive and encouraged Loehr [17] to make an attempt to implement it into a dialogue system, Elmo. During evaluation experiments, however, he realized that it is difficult to arrange generation of jokes that would be relevant to what users say, and that the genre of punning riddles is generally difficult to use in a dialogue [17].

Another attempt to create a practical application using JAPE was made by Ritchie et al. [28]. The algorithm of the generator was improved, and a user interface was added in order to make the interaction with humans easier to conduct. The target users of the system were children with CCN (complex communication needs), which can result in lower levels of literacy. The developed software (named STANDUP), was "a language playground, with which a child can explore sounds and meanings by making up jokes, with computer assistance" [28]. The results of evaluation experiments showed that the system was highly appreciated by the participants (children with CCN), and led to improvement of their communication skills.

This application is a very important step forward in the field of humor-equipped agents, as finally we could see a pun generator implemented into a system that actually interacts with humans. What is more important, the users appreciated this fact and evaluated the system as generally good and usable. This work gives us several important clues for our research – we know that humor can actually work well in human-agent interaction, and that creating applications that would be able to tell jokes during conversation is a worthwhile enterprise. Thus, in our research we are aiming to create a system which could be used also by healthy people, as a daily conversational partner.

The JAPE system was also converted into Japanese [3]. However, to our knowledge, the system (named BOKE) has not been implemented into any application that would interact with users.

Among other existing pun generators also worth mentioning is McKay's WISCRAIC system [19], which generated simple idiom-based witticisms, such as:

"The friendly gardener had thyme for the woman!" (thyme – type of a plant; homophone for the word "time" in the idiom "have time for someone"). The output is quite interesting, and importantly, it is also generated according to the context (not in a dialogue, but inside the sentence). The generator can be helpful for non-native English learners, as the presence of humor makes words and phrases (in this case – idioms) easier to remember. This is also an important idea regarding application, as humor is used here to help users achieve a particular task (learning a foreign language). Hopefully in the near future we will see WISCRAIC implemented into a system that interacts (converses) with humans in order to assist them with their study of idioms in English.

Another attempt at implementing a humor-generating engine into a chatterbot was made by Tinholt and Nijholt [36], who constructed a cross-reference joke generator and combined it with a dialogue system. The input for the generator is a sentence (utterance), which serves as a basis to produce humorous misunderstandings. For example, to a user's utterance:

User: Did you know that the cops arrested the demonstrators because they were violent?,

the system can generate a quasi-misunderstanding response:

System: The cops were violent? Or the demonstrators? :)

This concept of dialogue-integrated humor generation appears to be a useful idea. The evaluation of the system was performed by having it analyze several chat transcripts and a simple story text. The authors originally planned to conduct a user-focused experiment – however, they found it impossible, due to the fact that this type of cross-reference ambiguity occurs very rarely in real-life conversations [36]. This issue, however, does not necessarily exclude cross-reference joke generators from research on humor-equipped talking agents. Such a generator could be implemented into a chatterbot as one of several humorous modules, used only when the possibility of humorous misunderstanding would be detected.

All works described above aimed at building humor generators. However, there are also various studies in which humor is not generated automatically. Instead, human-created jokes are selected from a database or completely preprogrammed. Although such works do not contribute directly to the field of computational humor, they do prove that humor actually can enhance HCI.

One notable example of such studies was conducted by Augello et al. [1]. Jokes in their Alice-type chatterbot were not generated; the agent would ask the user if he/she wants to hear a joke and what it should be about, and then choose a proper joke from a database.

Such a setup is quite natural in Western cultures, where jokes are often announced before they are told – e.g. "I know a good joke, do you want to hear it?" This mechanism, however, does not work in Japanese, which is the language we are dealing with in our research. The puns in Japanese (called *dajare*) are usually inserted naturally into a conversation, in order to surprise the listener. Thus, the setup proposed by Augello et al. would presumably not be appropriate to Japanese.

The assumption that the benefits of humor should also work well in HCI was explicitly proved by Morkes et al. [21], who showed that a task-oriented conversational agent equipped with preprogrammed humor was evaluated as more sociable, likeable and easier to cooperate with by the users than a similar agent without humor.

To summarize this section, humor processing and its role in HCI is not an entirely neglected issue, as there are numerous works in this field. In most of these studies, utilizing benefits of humor by definition is associated with emotions (i.e., humor is used to enhance the interaction positively). However, to our knowledge, no existing research unites the two fields of humor processing and emotiveness detection. Consequently, no joke telling dialogue agent has been constructed that would be able to use humor deliberately, i.e. as a reaction to users' particular emotional states.

Thus, in this research we propose such a combination of humor and emotions. As we are aiming to construct agents that would interact with humans in a better and more natural way, we obviously want to enhance humans' feelings towards the agents. Including emotiveness detectors in such agents seems to us a logical progression in research on humor in HCI.

2. Our research

As mentioned above, the main goal of our research is to create a humor-equipped conversational agent, able to use humor appropriate to users' emotional states, in order to make them feel better. The research is still ongoing. In the following sections we present our achievements so far (Section 2.1) and the new contribution presented in this paper (Section 2.2).

2.1. Our research so far

In one of our previous works, we described an algorithm that – based on Dybala's complex classification of Japanese puns [6] – generates puns in Japanese, and added it into a chatterbot created by Higuchi et al. [11]. The reason we chose Japanese for the base language of our research is that in comparison to other languages it contains a large amount of homophones, which makes it easier to create puns.

The system we constructed generates joke-including answers using the interlocutors' utterances as input, in order to make them at least partially relevant to what the users say. Below we present an example of the system in action:

User: – *Kaeru daikirai!* (I hate frogs!)
System: – *Kaeru to ieba tsukaeru no desu ne.* (Speaking of frogs, we could use that!)

Evaluation experiments showed that the humor-equipped chatterbot was actually appreciated and found to be better than a non-humor-equipped one by users. Thus, it can be said that our system was successfully integrated into a normal conversation and worked robustly during evaluation experiments, in which the users were actually talking with the system. The presence of humor significantly improved users' opinions on the agent in all categories, such as willingness to continue the interaction or users' interest in the partner's talk [7].

As part of our research on an emotion recognition agent, we have also developed an automatic emotiveness-analysis-based method for dialogue systems evaluation. The chat logs from an experiment with our humor-equipped agents were analyzed using ML-Ask Emotive Elements/Emotive Expressions Analysis Agent, developed by Ptaszynski et al. [25,26]. ML-Ask analyzed users' emotive reactions towards both (humor- and non-humor-equipped) agents, searching for positive and negative emotions. The results showed that the agent with humor triggered much more positive reactions in users, while for the baseline agent without humor, the proportions were opposite [8].

In all previous experiments, we have used a very simple timing rule for the humor-equipped agent. It can be summarized as a "joke-at-every-third-turn" rule. This simple setup allowed us to check the influence of humor on the chatterbot's performance; however, it obviously needed to be changed. In this paper we present a solution to this problem (see below).

During experiments with the previous version of our system, we experienced some serious problems with the baseline chatterbot, caused by the fact that it uses Internet search engines (such as Google) in a very intensive manner (i.e., performs large numbers of queries). Thus, the agent would freeze every time our IP was recognized and blocked as a spam sender. To solve this problem, we have developed a new, simplified chatterbot – see Section 3.1.

2.2. Our contribution

The research described in this paper is novel and original in several ways. Firstly, to our knowledge, the multiagent joking system presented below (see Section 3.4) is the first conversational system able to tell jokes "deliberately" in reaction to users' emotional states. By "deliberately" we mean that the timing of jokes is no longer random (c.f. the "every-third-turn-rule" in the previous version of the system), and the system does not simply joke whenever possible. Instead, the decision whether humor should be used or not is based on the results of users' emotions (this role is performed by the ML-Ask agent). The fact that humor can make users feel better was confirmed in related works (see Section 1.3) as well as in our earlier studies (see Section 2.1) – however, in all of these cases, the focus of the research was laid on emotions following the jokes, i.e. on users' reactions to humor. We cannot forget, however, that humor in a conversation is also a reaction to something – users' emotions, context etc. In this research we propose a system which takes the former into consideration. Humor can be seen here as a method used by the system to make humans feel better.

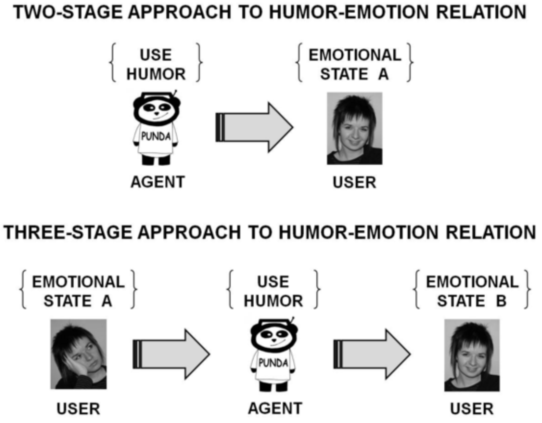

Fig. 1.Two- and three-stage approach to humor-emotion relation in human-agent interaction.

If we summarize the way humor was seen in earlier works (including our study) in relation to emotions, we can say that the approach was two-staged:

(1) System: use humor ⇒ (2) User: emotional state A (reaction to humor)

However, in this research we have broadened the approach, which now is three-staged:

(1) User: emotional state A ⇒ (2) System: use humor ⇒ (3) User: emotional state B (reaction to humor)

These two approaches are also illustrated in Fig. 1.

Stage (1), in which the emotiveness analysis of the user's utterance is conducted, is performed by the ML-Ask agent. Stage (2) is performed by the PUNDA pun generator if the decision is that a joke should be told; if the system decides otherwise, stage (2) is performed by the Maru-chan baseline chatterbot. Stage (3) was checked in the evaluation experiments (users' self-reported emotional states and chat log analysis by ML-Ask).

Also, as far as the agents are concerned, this research is the first in which the ML-Ask agent uses a web mining technique (see Section 3.3) to evaluate the system's performance. Some experiments in this area have been described earlier [27], but these were conducted on human-created sentences. Thus, in the experiment described in this paper (see Section 4.2) the ML-Ask agent uses the web mining procedure for the first time in an evaluation experiment of a conversational system.

This paper is also the international debut of the baseline chatterbot “Maru-chan” (see Section 3.1 for details). Earlier it was only described in Takahashi’s bachelor’s dissertation, which was in Japanese. This was also the first time when Maru-chan was used as a part of multi-agent system.

Also, in comparison to our previous works, the PUNDA pun-generating agent was further improved. Its previous version used only 4 pun generation patterns (homophony, initial mora addition, internal mora addition, and final mora addition). In the new version, used in this research, we implemented 3 new patterns: final mora omission, internal mora omission, and mora transformation) – see Section 3.2 for explanation. This significantly expanded the agent’s joke generation potential.

In the previous version of the PUNDA agent, we used an on-line sentence database to generate joke-including sentences. Parts of human-created sentences were utilized to create the system’s humorous answers – this, however, could impair the results of evaluation experiments, in which we are investigating issues such as human-likeness. Thus, in the new version, we have retired this idea and decided to use only templates of joke-including utterances (such as “speaking of [A], it’s [B]” – see Section 3.2 for details).

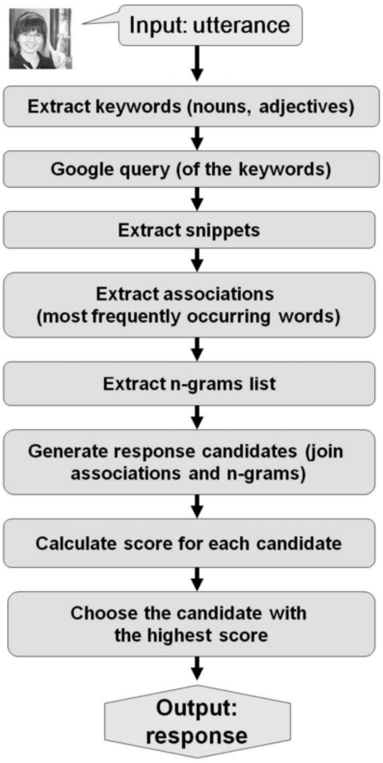
Input: utterance
Extract keywords (nouns, adjectives)
Google query (of the keywords)
Extract snippets
Extract associations (most frequently occurring words)
Extract n-grams list
Generate response candidates (join associations and n-grams)
Calculate score for each candidate
Choose the candidate with the highest score
Output: response

Fig. 2. Maru-chan (baseline chatterbot) algorithm outline.

3. The agents

The three agents used in this research are: a non-task-oriented conversational agent (chatterbot), a pun-generating agent (implemented into the chatterbot to construct a joking agent) and an emotiveness analysis agent. They were combined to cooperate in MAS-Punda, a multiagent joking conversational system.

3.1. Conversational agent (chatterbot)

As mentioned in Section 2, in our research so far we have used Higuchi et al.’s conversational agent Modalin [11] as a base to construct a joking chatterbot (also as a baseline system in our evaluation experiment). The agent performed relatively well and we were planning to keep using it in further research; however, we faced a serious problem. As Modalin used the Internet (search engines such as Google or Yahoo) as a database to extract word associations for the user’s utterance, for each turn of a dialogue, multiple queries had to be made in order to generate a proper response. This, in turn, led to overuse of search engines, which would recognize us as potential spam senders and block our IP. Problems like this caused us to cancel some of our evaluation experiments, as the chatterbot would freeze in the middle of interaction.

Currently, we are working on a corpus-based (non-Internet-using) version of Modalin. However, until it is finished, we decided to use another chatterbot, developed by Takahashi [34].

The agent, called Maru-chan, also uses the Internet as a source of linguistic knowledge, but – unlike Modalin – it performs only a limited amount of queries (usually 1 query per turn; Modalin – sometimes even above 100). Accordingly, we can avoid being blocked by search engines and the interaction with users can go smoothly.

The outline of Maru-chan’s algorithm is presented in Fig. 2.

In the first step, the agent extracts keywords from the user’s utterance (adjectives and nouns). For example, from the sentence:

“*Reizouko no naka ni tsumetai nomimono ga arimasu*” (There’s something cold to drink in the frigde), the extracted keywords would be "*reizouko*" (fridge), "*tsumetai*" (cold) and "*nomimono*" (something to drink).

Next, these keywords are used to perform a query in the Google search engine. In the case of the above example, the query line would be: "*reizouko tsumetai nomimono*". Then, snippets from the result pages are extracted.

In the next step, the agent uses the snippets to extract word associations (nouns and adjectives with the highest occurrence frequency), as well as n-grams (strings of words: sets of 3- and 4-grams for every word from the list) including these association words[1].

Next, these extracted n-grams and candidates are used to generate response candidates. If, for the above example, one of the association words was *juusu* (juice) and the one of the n-gram sets was: "*tsumetai*" (cold), "*juusu*" (juice), "*nara*" (if), "*sakki*" (just), "*nomimashita*" (drink – past final form) and "*yo*" (emphatic particle), the response generated would be "*tsumetai juusu nara sakki nomimashita yo*" (If you mean cold juice, I have just drunk it!).

After generating the response candidates, the agent gives a score to each of them. The criteria to do this are multiple – if, for example, the candidate includes one of the keywords extracted from the user's utterance, one point is added. Also, if the candidate forms a full sentence (ends with a final form of adjective or verb, or with a final particle), the agent adds one more point. For example, the example candidate "*tsumetai juusu nara sakki nomimashita yo*" (If you mean cold juice, I have just drunk it!) would get one point for including the keyword *tsumetai* (which was present also in the user's utterance) and one point for ending with the final past form *nomimashita* and a final particle *yo* (emphasis). Another criterion is the length of the candidate – the longest and the shortest candidates are automatically deleted (the former convey too little information, and the latter are often too complicated), and the medium candidates receive scores according to a manually set threshold (set empirically after a series of preliminary experiments).

The scores of all response candidates are compared and the top one is selected as the agent's response [34].

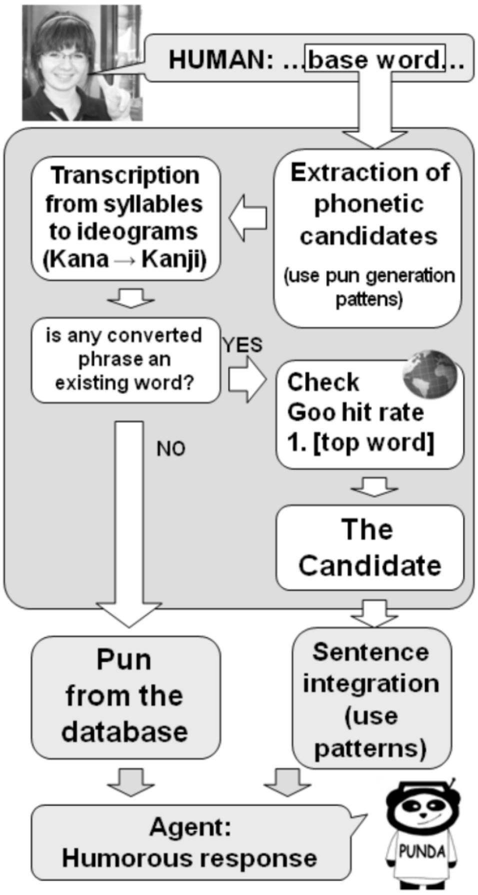
HUMAN: ...base word...
Transcription from syllables to ideograms (Kana → Kanji)
Extraction of phonetic candidates
(use pun generation pattens)
is any converted phrase an existing word?
YES
NO
Check Goo hit rate
1. [top word]
The Candidate
Pun from the database
Sentence integration (use patterns)
Agent: Humorous response
PUNDA

Fig. 3. PUNDA (pun generator) algorithm outline.

3.2. Pun generator

The pun generator (named PUNDA) also uses the Internet as a source of knowledge. Its input is an utterance (phrase or sentence) from which a pun base word (usually an ordinary noun or adjective) is extracted. Next, the system generates a list of phonetic candidates using Dybala's pun generation patterns (see below). The candidates are converted into *Kanji* (Chinese characters) and their hit-rates are checked on the Internet. The candidate with the highest hit-rate is selected and integrated into a sentence. As output, the system produces a joke-including response.

An outline of the algorithm is presented in Fig. 3.

In our research, we based our work on a complex Japanese pun classification proposed by Dybala [6]. The puns were divided into 12 groups (with numerous subgroups), according to mora (phonotactical unit, in most cases equal to a syllable) changes be-

[1] In general, this n-gram model seems usable also in other languages, with small differences, such as the value of n (which may be higher or lower, depending on the type of language). In the example with juice, an extracted set of n-grams might look like this: "I" "have" "bought" "a" "cold" "juice" (6-gram).

tween the base phrases and phrases transformed into a pun. For example, in a simple pun "*kono kusa wa kusai*" ("this grass stinks"), the base phrase "*kusa*" (grass) is transformed into "*kusai*" (to stink), and the technique used is called "final mora addition", as there is one mora ("*i*") added to the end of the base phrase.

The classification was used to create pun generation patterns (an equivalent of JAPE's schemata). For example, the group called "final mora addition" produces a pattern that can be transcribed as [base phrase][*], where [*] means one single mora. Currently, there are seven patterns implemented in the system – presented below, with the word *katana* (a Japanese saber) as an example:

1. homophony ([*katana*])
2. initial mora addition (**katana*: *akatana, ikatana, ukatana...*)
3. internal mora addition (*ka*tana, kata*na: kataana, kataina, katauna...*)
4. final mora addition (*katana*: katanaa, katanai, katanau...*)
5. final mora omission (*kata*)
6. internal mora omission (*kana*)
7. mora transformation (*gatana, tatana, matana...*).

In pattern 7 the number of possible transformations is very large (assuming that any sound can be transformed into any other sound). Therefore, in our system we used Japanese phoneme similarity values, proposed by Takizawa et al. [35] to find phrases that sound more similar than others.

The patterns were used in our pun candidate generation algorithm, in order to generate phonetic candidates, as showed above for the base word *katana* (see also Fig. 3). In the next step, each phonetic candidate is converted to *Kanji* (Chinese characters). If there is more than one possible conversion, all options are extracted. Next, all converted phrases are used as query words on the Internet (currently using Yahoo's search engine), and the phrase with the highest hit rate is selected as the final candidate for a pun.

Finally, the candidate is integrated into a sentence. To do this, we extracted some general templates used in pun-including conversations between humans. For example, one such pattern is:

[base word] *to ieba* [pun candidate] *desu ne.*
(Speaking of [base word], it's [pun candidate]).

So, as output, the system produces a pun-including response, in which the base word is repeated (in order to mark the relevance to the previous utterance and to make the pun more visible) and the pun candidate is presented in one sentence.

In comparison to the system's previous version, we have added a new pun generation pattern (mora transformation). We have also abandoned the semi-automatic sentence integration algorithm; in the previous version of the system, pun-including utterances were formed using parts of human-created sentences, extracted from an on-line corpus. We have replaced this option with automatically generated templates (see above).

3.3. Emotiveness analysis agent (ML-Ask)

Another agent used in this research is Ptaszynski et al.'s ML-Ask Emotive Elements/Emotive Expressions Analysis System [25,26]. As mentioned above, in this system the emotiveness analysis agent performs two functions:

1. it decides if it is appropriate to tell a joke (see Section 3.4), and
2. it performs automatic evaluation of the results (see Section 4.2).

ML-Ask's algorithm is based on Ptaszynski's idea of binary classification of realizations of emotions in language [24].

Ptaszynski [24] distinguished two kinds of realizations of emotions in Japanese: emotive elements, which indicate that emotions have been conveyed, but not detailing their specificity (this group is linguistically realized by interjections, exclamations, mimetic expressions, or vulgar language) and emotive expressions – parts of speech like nouns, verbs, adjectives or metaphors describing affective states.

One of the assumptions of Ptaszynski's binary classification is that to contain any specified emotions, the sentence must first contain emotive elements. Thus, even if a sentence contains emotive expressions, it does not necessarily mean that it is emotive and contains emotions. For example, in the sentence:

Ryoushin wa minna jibun no kodomo wo aishiteiru.
(All parents love their children),

we can find an emotive verb *aishiteiru* (to love), but the sentence is a generic statement and, if not put in a specific context, does not convey any emotions. Therefore, the ML-Ask agent only specifies types of conveyed emotions if there are any emotive elements in the utterance.

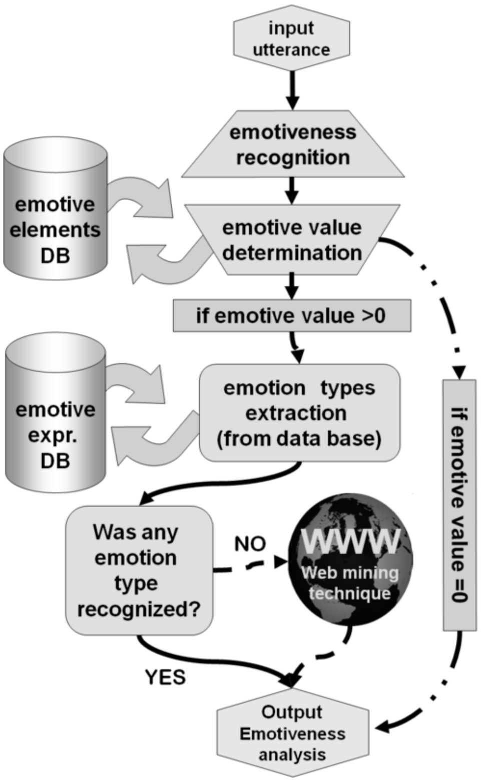

Fig. 4. ML-Ask (emotiveness recognition agent) algorithm outline.

The ML-Ask system performs utterance analysis in three general steps:

1. Determining general emotiveness (if the utterance is emotive/non-emotive)
2. Specifying types of emotions found (in emotive utterances only)
3. Specifying valence (positive/negative).

Descriptions of these steps are presented in the following sections.

3.3.1. Determining general emotiveness

In the first step, the agent performs analysis in order to check for the presence/absence of emotions and determines their emotive value in utterances. For example, the sentence:

> "*Kono hon saa, sugee kowakatta yo. Maji kowasugi!*" (That book, ya know, 'twas a total killer. It was just too scary.),

is recognized as emotive, as it contains emotive elements: *saa* (emphasis), *sugee* (totally), *yo* (emphasis), *maji* (really), -*sugi* (too much) and an exclamation mark. Emotive elements do not belong to any particular type of emotions, but make the utterance more emotive. Summing these up, in the first step of analysis the above sentence is denoted as: emotive, with emotive value = 6 (total amount of emotive elements).

3.3.2. Specifying types of emotions

In the second step, if the emotive value (detected in the first step) is higher than zero (i.e. there are emotive elements present in the utterance), ML-Ask starts searching for specific types of emotions. To do this, it uses a database created on the basis of Nakamura's Japanese emotions classifications (10 types) [22]. First, ML-Ask checks if any of words found in the utterance can be found in the database. If yes, it extracts these words (emotive expressions) and as output produces the expression(s) and emotion type(s) to which they belong. For example, in the sentence above, the agent found the emotive expression *kowai* (scary), which belongs to the group called *kyoufu* (fear).

However, there are cases where a sentence is emotive (checked in the first step), but does not include any of the emotive expressions from the database. For example, in the sentence:

> "*Kyou wa atatakai desu ne*." (It's warm today, isn't it?),

the agent finds the emotive element "*ne*" ("isn't it"), but does not detect any particular emotions.

When such a situation occurs, ML-Ask uses Shi et al.'s [38] web mining technique as a support method for extracting emotive contents.

Web mining technique

Nowadays, the Internet is becoming not only a source of encyclopedic information, but also of other types of knowledge, such as human commonsense [30]. People describe their experiences and feelings on blogs, social networks and other types of domains. This opens new, exciting possibilities for the field of information retrieval, as all these blocks of knowledge in the Web can be extracted and analyzed. The Internet is easily available, up-to-date and is expanding every second; it is a huge database that can be used to acquire many types of information, including those concerning human emotions and affects. During the last few years we have seen some (albeit few) attempts at using web mining in the field of affect analysis (e.g. [10,18]). In this research we use the web mining technique for Japanese, proposed by Shi et al. [38].

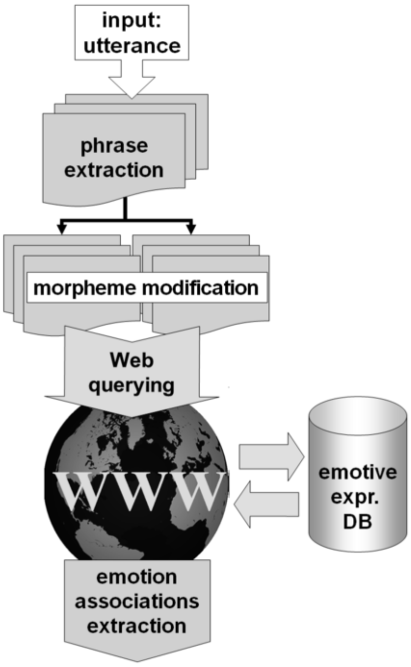

Fig. 5. Shi's web mining technique (part of ML-Ask algorithm).

The web mining technique consists of three steps: a) phrase extraction from an utterance; b) morpheme modification; and c) extraction of emotion associations.

a) Phrase extraction

An utterance is first processed by MeCab, a tool for part-of-speech analysis of Japanese [13]. Every element separated by MeCab is treated as a unigram. All unigrams are grouped into larger n-gram groups preserving their word order in the utterance. The groups are arranged from the longest n-gram (the whole sentence) down to all groups of trigrams.

Thus, in the case of the above example, the phrases would be: "*kyou wa atatakai desu ne*" (the whole sentence – 5-gram), "*kyou wa atatakai desu*", "*wa atatakai desu ne*" (4-grams), "*kyou wa atatakai*" and "*atatakai desu ne*" (3-grams).

After a series of preliminary experiments, n-grams ending with particles were excluded, since they gave too many ambiguous results. Thus, in the above example, all phrases ending with *ne* are deleted from the list. After this step, there are two phrases left: "*kyou wa atatakai desu*" and "*kyou wa atatakai*".

b) Morpheme modification

For semantically deeper Web mining, after extracting a list of phrases from the utterance, all n-grams ending with a verb or an adjective are grammatically modified in line with Yamashita's research on causality morphemes, after which people tend to convey emotive meaning in Japanese [39]. This research was also experimentally confirmed by Shi et al. [38], who distinguished eleven emotively stigmatized morphemes for the Japanese language using statistical analysis of the Web contents. Next, they used the Internet to check which of these eleven causality morphemes were most frequently used to express emotions (those from Nakamura's emotive expressions data base). On this basis, they chose five morphemes with the highest frequency to be used in the process of n-grams modification. The causality morphemes are: -*te*, -*to*, -*node*, -*kara* and -*tara* (see Table 1).

Table 1
Hit-rate results for each of the 11 morphemes (5 with the highest rates were chosen to be used in the procedure)

morpheme	*-te*	*-kara*	*-nara*	*-kotowa*
result	41.97%	6.32%	1.17%	0.30%
morpheme	*-to*	*-tara*	*-noga*	*-nowa*
result	31.97%	5.94%	2.15%	2.30%
morpheme	*-node*	*-ba*	*-kotoga*	
result	7.20%	3.19%	0.35%	

Thus, for one of the n-grams that passed the selection in the previous step: -*kyou wa atatakai* (it's hot today), the phrases after morpheme modification would be:

kyou wa atatakakute (it's hot today, and-)
kyou wa atatakai to (if it's hot today-)
kyou wa atatakai node (because it's hot today-)
kyou wa atatakai kara (because it's hot today-)
kyou wa atatakattara (if it's hot today-)

c) Emotion type extraction

All the modified phrases acquired in the previous phase are used as query words in Yahoo's search engine. 100 snippets for each query phrase are extracted and cross-referenced with the database of emotional expressions. The emotive expressions extracted from the snippets are summed up, and the results for every emotion type are listed in descending order. This way, a list of emotions commonsensically associated with the queried sentence is obtained (results for the "*Kyou wa atatakai desu ne*" example are showed in Section 3.3.3).

However, emotive associations extracted from the Web contain a certain amount of noise. Ptaszynski et al. [26] showed that Shi's technique is the most efficient when only the emotions with the highest hit rate are kept and the rest is considered as noise.

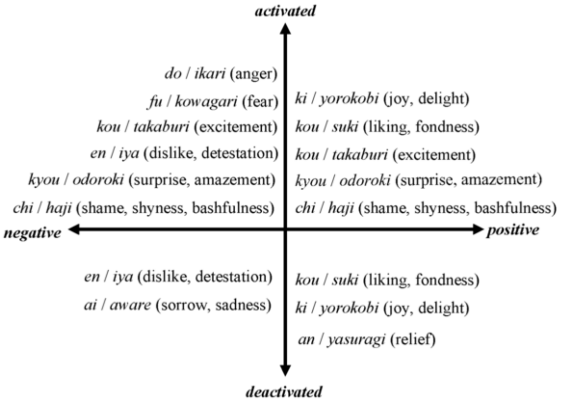
activated
do / ikari (anger)
fu / kowagari (fear)
kou / takaburi (excitement)
en / iya (dislike, detestation)
kyou / odoroki (surprise, amazement)
chi / haji (shame, shyness, bashfulness)
ki / yorokobi (joy, delight)
kou / suki (liking, fondness)
kou / takaburi (excitement)
kyou / odoroki (surprise, amazement)
chi / haji (shame, shyness, bashfulness)
negative
positive
en / iya (dislike, detestation)
ai / aware (sorrow, sadness)
kou / suki (liking, fondness)
ki / yorokobi (joy, delight)
an / yasuragi (relief)
deactivated

Fig. 6. Grouping Nakamura's classification of emotions on Russell's two-dimensional space.

3.3.3. Valence specification

In the third step, the results acquired in the procedures described above are summarized, and the sentence's emotive valence is specified. Nakamura's types of emotions have been divided into positive and negative. To do that, we used Russell's 2-dimensional model of affect [29], translated into Japanese. The results of projecting Nakamura's emotion types on the Russell's model are showed in Fig. 6.

The main assumption of this idea is that all emotions can be described in a space of two-dimensions: the emotions' polarity (positive/negative) and activation (activated/deactivated). In Fig. 6, some types were placed in two quarters, as they can contain both positive and negative or activated and deactivated expressions. This, however, only concerns types (groups) of emotions – each of the emotive expressions belongs only to one group.

If an emotive expression found in a sentence belongs to the positive group, it is counted as positive, and if to the negative group, as negative. The dimension of activation/deactivation is not taken into consideration in this particular research – however, we are planning to use it in the future.

Below we summarize the results of analysis for example sentences used in above sections:

Sentence: *Kono hon saa, sugee kowakatta yo. Maji kowasugi!* (That book, ya know, 'twas a total killer. It was just too scary.)

Emotive elements: *saa* (emphasis), *sugee* (totally), *yo* (emphasis), *maji* (really), -*sugi* (too much), exclamation mark

Emotive value: 6 (above zero ⇒ specify types of emotions)

Emotive expressions: *kowai* (frightening)

Emotions found: fear

Valence: negative

Sentence: *Kyou wa atatakai desu ne.* (It's warm today, isn't it?)

Emotive elements: -*ne* (-isn't it)

Emotive value: 1 (above zero ⇒ specify types of emotions)

Emotive expressions: none (⇒use web mining procedure)

Emotions found on the Web: joy

Valence: positive

The ML-Ask Emotiveness Analysis Agent in our joking system performs the role of a "judge", deciding if it is appropriate to tell a joke (see Section 3.4). It was also used in automatic evaluation of the system (see Section 4.2).

3.4. MAS-Punda: Multiagent joking system

The agents presented in above sections were joined together to cooperate in MAS-Punda, a humor-equipped joking conversational system.

As mentioned above, in our previous research we used a very simple "joke-at-every-third-turn" rule. In this research – and herein lies one of the novel ideas of this work – the role of timing decision maker was played by the ML-Ask agent.

As presented in Section 1.2, humor can help us deal with negative moods, as it holds the power to turn them into positive ones. Thus, we assumed that in an interaction between humans and computer agents, the latter could use humor to make the interlocutor feel better.

In order to do that, first the system has to detect the human partner's emotions. This function in our joking system is performed by the ML-Ask agent.

As in some cases the web mining procedure is quite time-consuming, in the emotion recognition process during conversations with humans we abandoned this option and used the database-only emotion specification pattern (in the automatic evaluation experiment, however, we used both database and web mining procedures).

Based on the findings described in Section 1.2, we assumed that:

- if the human's emotive state is negative, the agent can use humor in order to make him / her feel better / reduce the negativity

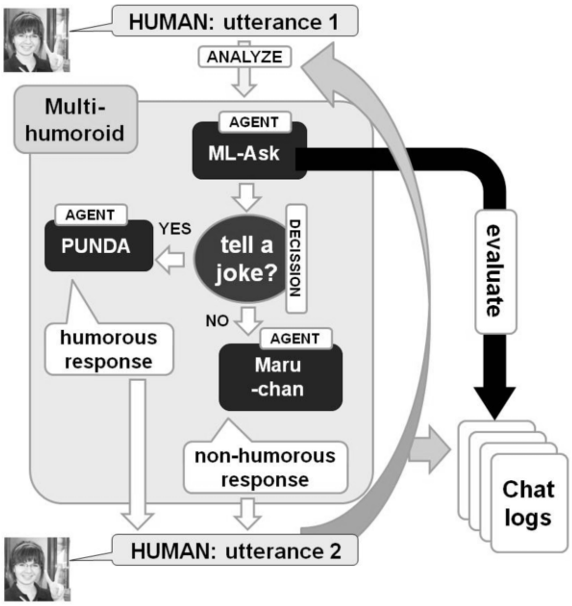

Fig. 7. Multiagent joking system (MAS-Punda) algorithm outline.

- if the human's emotive state is neutral (non-emotive), the agent can use humor to induce positive emotions.

As we wanted to study user reactions to the agent's performance, we decided that jokes should not be told when the user's utterance is emotive but no particular types of emotions are detected, as it would be hard to analyze how users' emotive states changed when we do not know the initial state.

Thus, the decision patterns for the ML-Ask agent were:

1) if user's emotive state is negative ⇒ tell a joke
2) if user's emotive state is neutral ⇒ tell a joke
3) otherwise – do not tell a joke

If the ML-Ask agent decides that it is appropriate to tell a joke, a response to the user's utterance is generated by the PUNDA (joke generating) agent. If ML-Ask decides otherwise, the response is generated by the Maru-chan.

The outline of the multiagent joking system is presented in Fig. 7.

4. Evaluation experiments

To evaluate the effects of our work on the multiagent joking system, we conducted two experiments: a first-person (user) oriented (conversations with two agents), and automatic (emotiveness-analysis-based).

4.1. First person (user) oriented evaluation

In the first experiment, we asked 13 university students (age 21–30) to perform two conversations: one with Maru-chan (the baseline chatterbot) and one with the MAS-Punda (the multiagent joking system). The order of conversations was randomized. There was no topic restriction, thus the conversations could be about any subject. All interactions were text-based.

The links to on-line applications were sent to the participants by e-mails, so the conversations could be performed wherever they had access to the Internet. The participants were asked to conduct the interactions continuously, i.e. one right after another (in order to make the comparison easier).

The participants also received links to questionnaires (filled in on-line), including questions concerning the interaction.

The questions were:

A) Did you get an impression that the agent was human like?
B) Did you get an impression that the agent tried to make the conversation more interesting?
C) Did you find the conversation interesting?
D) Did you get an impression that the agent tried to make your feelings better / more positive?
E) Do you think that the agent used humor in appropriate moments?
F) Please describe your feelings towards the agent after the interaction
G) If you were to make friends with one of these agents, which would you choose?
H) Which agent do you think was better?

Answers for questions A–E were given in 5-point scales. For question F, the evaluators could answer freely (they could write whatever they wanted). For questions G and H, the evaluators had to choose between the two agents.

As only one of the agents used humor (intentionally), answers for question E (directly related to humor) also included option "the agent did not use any humor".

The results for question F were compared with our emotive expressions data base, in order to check the valence of each emotion listed by users. Every positive emotion counted as +1, and every negative as –1. All results for each agent were summarized (see Table 2).

Table 2
User-oriented evaluation experiment – results

Question	Maru-chan	MAS-Punda	Difference	P value
A	1.85	2.15	0.30	0.090
B	1.69	2.85	1.16	0.006
C	2.08	2.69	0.61	0.026
D	1.69	2.69	1.00	0.007
E	1	2.45	1.45	0.001
F	–9	+8		
G	30.8%	69.2%		
H	38.5%	61.5%		
continue?	2	5		

In our previous experiments, we also asked the users if they wanted to continue the interaction. In this research, we decided to check this in a more direct way; after ten turns of conversation, the agents asked the users if they wanted to continue the interaction. If the answer was "yes", another five turns of conversation were performed. After that, the conversation ended automatically (similarly if the answer was "no").

The results for the questions were summarized and the statistical significance was calculated. For questions A–D, we used the Two Paired Sample Wilcoxon Signed Rank Test (as the data was paired and not distributed normally). For question D, we used the Mann–Whitney U test (as the data was unpaired and not distributed normally).

4.2. Automatic evaluation

The chat logs acquired in the user experiments were next analyzed with the ML-Ask emotiveness analysis agent. Results of the analysis allowed us to compare the dialogues of our two systems (with and without humor) in two aspects:

1) General emotiveness
(Calculated by summarizing emotive value for all users' utterances from conversations with each agent)
2) Valence changes (positivity/negativity)
(If emotions detected by ML-Ask changed from negative or neutral to positive, the change was assumed to be positive; if the change was from positive or neutral to negative, the change was assumed to be negative)

Regarding the merit of conducting automatic affect analysis, in our research we decided to apply the "affect-as-information" approach, proposed by Schwarz and Clore [31]. Its main idea is based on the claim that humans use affect in the same way as other criteria, namely to form their opinions and judgments, based on the informational value of their affective reactions. This leads to the assumption that information about someone's attitude to a product can be derived from information about changes in his or her affective states during its usage [12]. Thus, the analysis of user feelings towards agents should allow us to investigate their judgments and opinions about them.

The results of both experiments are summarized below.

5. Results

The majority of the results of both experiments are satisfying. The multiagent joking system was evaluated as generally better than the baseline (non-humor-equipped) agent.

The results of both (user-oriented and automatic) experiments are presented in Sections 5.1 and 5.2.

5.1. First person (user) oriented evaluation

In all categories MAS-Punda received higher scores than Maru-chan. However, not all of the results were statistically significant. The summary is presented in Table 2.

As shown above, the smallest (and not significant) difference was observed in the responses to question A (regarding the agents' human-likeness). We can still see that there is a tendency favoring the agent with humor; however, the lack of statistical significance prevents us from drawing any unanimous conclusions. This may be due to the vague nature of the concept of computers being human-like, as obviously the current level of science is still quite far from that level.

A difference between the two agents is clearly visible in the results of question B (Did the agent try to make the conversation more interesting?). This signifies that the presence of humor in MAS-Punda's responses was appreciated as an effort in order to enhance users' interest in the interaction. Of course, here we face an important question: what were the results of these efforts? The answer lies in the results for question C (Was the conversation interesting?). The differences here are significant, which means that most efforts to make the dialogue interesting were actually successful.

The difference was also clear and significant in question D's responses (Did the agent try to make your feelings better / more positive?). This gives us very important information – that jokes used by the humor-equipped agent were recognized as attempts to make the partner feel better. Also, here we can question the efficiency of these efforts; the answer can be found in the results for question F (see below) and automatic evaluation (see Section 5.2).

As mentioned above, the answer options for question E (Did the agent use humor in appropriate moments?) included the option “the agent did not use humor”. In case of Maru-chan, five such answers appeared, while there were only two for MAS-Punda. These answers were not taken into consideration when calculating the score for this question.

As described in Section 4.1, the results for question F (Please describe your feelings towards the agent after the interaction) were analyzed by comparing them to our emotive expressions data base in order to check how positive/negative the user’s feelings after the interactions were. Every positive emotion counted as +1, and every negative as –1. After summarizing all scores for each agent, it was found that the agent without humor received an overall note of –9 (5 positive, 14 negative), while for the agent with humor the score was +8 (15 positive, 7 negative). This means that the agent with humor triggered more positive and less negative self-reported emotions in users. These results also give us the answer to the above question, namely, were the joking agent’s efforts to make the interlocutors’ feelings better successful? The answer is obviously: yes, in most cases they were.

The differences in the results of questions G (If you were to make friends with one of these agents, which would you chose?) and H (Which agent do you think was better?) are quite visible, although they could be better. However, almost 70% of users would still prefer the agent with humor as a friend, and more than 60% said that it was generally better than the one without humor.

Also, more users (five) decided to continue the interaction with the humor-equipped agent. One could point out that it is only five out of thirteen (38.5%) – however, this is still more than twice as many as for the non-humor-equipped agent (two persons, 15.4%). It is also worth mentioning that none of the users decided to continue the interaction only with the agent without humor.

5.2. Automatic evaluation

As mentioned above, the chat logs from the user experiments were analyzed by the ML-Ask agent. The analysis was conducted in two aspects: 1) general emotiveness (sum of all emotive values of all users for each system) and 2) valence changes (if emotions detected by ML-Ask changed from negative or neutral to positive, the change was assumed to be positive; if the change was from positive or neutral to negative, the change was assumed to be negative).

Table 3
Automatic (emotiveness analysis based) evaluation experiment – results

	Maru-chan		MAS-Punda	
general emotiveness	91 (average: 7.0 per utterance)		125 (average 9.6 per utterance)	
valence changes	to positive	to negative	to positive	to negative
	68%	32%	93.75%	6.25%

In both of these categories, the results indicate the superiority of MAS-Punda – it generally triggered more emotions in users (sum of emotive values = 125, comparing to 91 for Maru-chan), and more valence changes were positive (93.75% for MAS-Punda vs. 68% for Maru-chan).

The results are summarized in Table 3.

The implications of these results are discussed in Section 6.

6. Discussion

The results of both experiments generally confirmed our expectations. The MAS-Punda (multiagent joking system) was evaluated as generally better and tending to be more human-like than the baseline agent Maru-chan. The users also appreciated MAS-Punda’s efforts to make them feel better and to make the conversation more interesting. Moreover, in both of these cases, the efforts were relatively successful. Also, more users decided to continue the interaction with MAS-Punda, and more of them would choose this agent for a friend.

Some implications of these results are discussed below.

6.1. Subjectivity of the user evaluation

It may be claimed that the results of our user-oriented experiment do not give us any clear data, as they are not objective. We agree; the subjectivity of these results is obvious. The question, however, is: is this really a drawback?

In our research, we assume that ultimately it is the user who has to be satisfied. Thus, the user’s impressions (which are by definition subjective) about the product (in this case, an agent) are of top priority, and asking about them directly, as we did in this experiment, seems to us the most effective way to proceed.

From this point of view, the experiment was reasonably successful. The users evaluated the agent with humor as more interesting, making them feel more positively and slightly more human-like (of low statistical significance – we can only talk about a trend here).

Questions G and H may appear too general; however, if we are to investigate impressions of users as final clients of our "product", we have to think in a way clients do when they decide what to buy. If a client has to choose between two more or less similar products, everything comes down to the general question: which of the two is better?

For this reason, we decided to ask the participants the same question (H). Also, as we are aiming to create an agent that would be able to act as a human's talking companion, we wanted to investigate the potential friendliness of both agents (question G). The results for both of these questions indicate the MAS-Punda to be superior (69.2% for question G and 61.5% for question H) – however, the differences here are not as great as we expected. Some possible reasons for this are discussed in Section 6.2.

6.2. Individual differences

The fact that the differences in results of questions G and H were not as significant as they could be (four users chose Maru-chan for a friend and five evaluated it as generally better) was somewhat below our expectations. The tendency is still visible, and also here the agent with humor was assessed higher than the baseline one, but the proportions were not remarkably different.

However, we have to remember that we are dealing here with sense of humor – a trait of personality that can be completely different from individual to individual. What one person finds very funny, another person may find not funny at all. Some people may like a certain type of joke, whereas others may actually hate it.

The fact that MAS-Punda tells only puns, means that a user who does not like this type of humor would not appreciate the agent's performance. Also, the puns generated by MAS-Punda were rather simplistic, which may also have influenced the results.

The above, though, are only our speculations. In order to investigate these issues, before conducting future experiments we will have to perform a type of self-report test relating to sense of humor (as, for example, Svebak's Sense of Humor Questionnaire [32]) in order to determine what type of humor each user prefers. Acquired data could be compared with the results of the experiment, which could provide us with information about which of the users' answers were actually caused by their sense of humor preferences.

6.3. Users' emotive states and their changes

In this experiment we were particularly interested in the users' mood changes, as we are aiming to construct a conversational agent that would be able to make its human interlocutors feel better. We know that humor holds the power to change our moods (see Section 1.2), and the experiments described above showed that it worked well in our joking agent.

In order to verify users' changes of moods and feelings towards both agents, first we asked them to report these immediately after the conversation (question F). Here, the differences were clearly visible, and it can be said that the agent with humor generally made the users feel better, while for the baseline agent the feelings were generally negative.

One problem with having humans self-report their feelings is that they may not be fully aware of them. As mentioned above, the users' opinion is of the highest priority for us; however, in some cases they may not be aware of all their feelings. Some of these feelings, though, may be reflected in the textual layer of speech – which is why we decided to also conduct the automatic emotiveness analysis.

We also wanted to study changes in users' emotive states during the conversations, not only after the interactions. However, asking the experiment participants to self-report their mood changes after each turn seemed to us too troublesome (the participants were volunteers). It would also disturb the flow of interaction, and we wanted it to be as natural as possible. Therefore, we decided not to ask the users about their mood changes directly. Instead, we rely on the results of automatic emotiveness analysis, conducted by the ML-Ask agent, which show clearly that the agent with humor triggered much more positive mood changes in users (93.75%) than the baseline agent without humor (68%). Of course, also in the case of Maru-chan, there were more positive changes than negative; however, when we compare the results, we can see the difference, indicating MAS-Punda as the agent triggering much more positive changes of the users' feelings.

6.4. Timing of humor

As mentioned above, in previous experiments we used a very simple ("joke-at-every-third-turn") timing rule. In this experiment, the role of the "timing judge" was performed by the ML-Ask agent.

Question E in the user-focused evaluation was intended to investigate this issue. As one of the agents was non-humor-equipped, the answers list included the option "the agent did not use humor". In the case of Maru-chan, this option was chosen by five users, and in case of MAS-Punda, by two of them. This, however, may have been caused by problems relating to definition of humor in Japanese. In fact, in one of our previous experiments, we asked the participants if *dajare* (Japanese puns) are jokes and if they represent humor. Results showed that there was no clear tendency in the answers; some people claimed that puns are jokes, but that they do not represent humor, while others said that puns are humorous, but cannot be called "jokes". Thus, it seems that there is a problem with definition in the field of humor in Japanese – and this, in fact, could be the cause of two users claiming that the MAS-Punda did not use humor.

However, we also checked these two chat logs and counted the attempts at telling puns by MAS-Punda. During conversations with these two users, the agent made only one attempt to joke, while in general the average amount of such attempts per conversation was 3.38. Also, in both of these cases the quality of output was not very high, and may not have been recognized as a joke.

For question E, the responses which stated that the agent did not use any humor were not taken into consideration when calculating the average score (see Table 2). The average score for Maru-chan was 1.0. This may mean that even if some of its utterances did include some (unintentional) humor, its timing was evaluated as inappropriate. Contrary to that, in the case of MAS-Punda, the average score for timing was 2.45. Thus, it can be said that the timings of jokes (i.e. decisions made by ML-Ask) were moderately appropriate, and that the emotiveness-analysis-based timing algorithm is at least a step into the right direction.

Also here, however, we can ask ourselves whether the results could be better. The average of 2.45 is still quite far from 5, and improvement of this result would certainly be desirable. Some ideas of how to achieve this goal are presented in Section 7.

7. Conclusions and future work

Needless to say, the results of our evaluation experiments could be higher, and our algorithms still need to be improved. In the following sections we present some ideas.

7.1. Better timing

Although having the ML-Ask agent decide whether it is appropriate to tell a joke seems to have worked reasonably well, we are aware that its settings (joke if the emotive state is neutral/negative) are still too general. Thus, we need to specify which emotion types in particular are appropriate to be responded with humor, and which are not. To do that, we are planning to:

- build a pun-including human-human conversation corpus;
- have professional comedians construct a corpus of conversations including ill-timed humor;
- analyze these two corpora with ML-Ask to search for regularities;
- have the same corpus annotated by humans.

Having accomplished these steps, we will hopefully be able to specify which particular emotive states are, and which are not appropriate to use humor. In terms of common sense we know that humans do not use humor as a response to some emotion types, as, for instance, grief after someone's death. Specifying exactly which emotions (and possibly, which particular expressions) can be answered with humor, will allow us to create a set of rules that could be used in the timing algorithm (e.g. "if [grief] do not tell jokes").

7.2. Individualization of humor

In Section 6.1 we mentioned that the user-focused evaluation is by definition subjective, but in our opinion this is not necessarily a drawback. However, this subjectivity of evaluation could lead to a situation where, even if we constructed a very sophisticated system that is evaluated highly by most evaluators, we still cannot be sure that all users will like it, as there are individual differences that influence their assessment.

The best method to prevent such situations is to construct a system that would adapt to the user's needs. In our research we focus on the role of humor in conversation. Currently we are working on an emotiveness-analysis-based evolution of the humor algorithm [8], which will allow the system to check user reactions to particular jokes (using the ML-Ask agent), and on this basis build his/her sense of humor model. For example, if the user reacts with positive emotions to jokes concerning politics, the system can assume that this type of joke matches his/her sense of humor. In this manner, the longer the system talks to the user, the more accurate "tags" of sense of humor it could attach – and this, in effect, would lead to more personalized, more individualized jokes with a high probability of being appreciated by the user.

Acknowledgements

The research presented in this paper was partially sponsored by a research grant from the Nissan Research Foundation.

References

[1] A. Augello, G. Saccone, S. Gaglio and G. Pilato, Humorist Bot: Bringing Computational Humour in a Chat-Bot System, Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems 2008 (CISIS 2008). Barcelona, Spain, 2008, pp. 703–708.

[2] K. Binsted, Machine humour: An implemented model of puns, Ph.D. Dissertation, University of Edinburgh, UK, 1996.

[3] K. Binsted and O. Takizawa, Computer generation of puns in Japanese. Sony Computer Science Lab, Communications Research Lab, 1997.

[4] G. Castellano and P.W. McOwan, Analysis of affective cues in human-robot interaction: a multi-level approach, 10th International Workshop on Image Analysis for Multimedia Interactive Service (WIAMIS 2009), London, UK, 2009, pp. 258–261.

[5] R.A. Dienstbier, The impact of humor on energy, tension, task choices, and attributions: Exploring hypotheses from toughness theory, Motivation and Emotion, 19(4), 1995, pp. 255–267.

[6] P. Dybala, *Dajare – Nihongo ni okeru dōon'igi ni motozuku gengo yūgi* (Dajare – Japanese puns based on homophony). M.A. Dissertation, Jagiellonian University, Kraków, Poland, 2006.

[7] P. Dybala, M. Ptaszynski, S. Higuchi, R. Rzepka and K. Araki, Humor Prevails! – Implementing a Joke Generator into a Conversational System, Proceedings of the 21st Australasian Joint Conference on AI (AI-08), Wobcke, W. and Zhang, M., eds., Auckland, New Zealand, 2008. Springer-Verlag LNAI Vol. 5360 (2008), Springer Berlin & Heidelberg, pp. 214–225.

[8] P. Dybala, M. Ptaszynski, R. Rzepka and K. Araki, Humorized Computational Intelligence – towards User-Adapted Systems with a Sense of Humor, Proceedings of the EvoStar 2009 Conference, EvoWorkshops. M. Giacobini et al., eds., Springer-Verlag LNCS, Vol. 5484 (2009), Springer Berlin & Heidelberg, pp. 452–461.

[9] P.S. Fry, Perfectionism, humor and optimism as moderators of health outcomes and determinants of coping styles of women executives, Genetic, Social & General Psychology Monographs, 121(2), 1995, pp. 211–245.

[10] G. Grefenstette, Y. Qu and J.G. Shanahan, Coupling niche browsers and affect analysis for an opinion mining application, Proceedings of RIAO, 2004.

[11] S. Higuchi, R. Rzepka and K. Araki, A Casual Conversation System Using Modality and Word Associations Retrieved from the Web, Proceedings of EMNLP '08, Honolulu, USA, 2008, pp. 382–390.

[12] J. Jiao, Q. Xu and J. Du, Affective Human Factors Design with Ambient Intelligence, HAAI'07, 2007, pp. 45–58.

[13] T. Kudo, MeCab: Yet Another Part-of-Speech and Morphological Analyzer. 2001. http://mecab.sourceforge.net/.

[14] H.M. Lefcourt, K. Davidson, R. Shepherd, M. Phillips, K.M. Prkachin and D.E. Mills, Perspective-taking humor: Accounting for stress moderation, Journal of Social & Clinical Psychology, 14(4), 1995, pp. 373–391.

[15] I. Leite, A. Pereira, C. Martinho and A. Paiva, Are emotional robots more fun to play with?, Proceedings of the Robot and Human Interactive Communication RO-MAN 2008, The 17th IEEE International Symposium, Munich, Germany, August 2008, pp. 77–82.

[16] LIREC – LIving with Robots and InteractivE Companions, http://www.lirec.org/.

[17] D. Loehr, An integration of a pun generator with a natural language robot, Proceedings of the International Workshop on Computational Humor, J. Hulstijn, A. Nijholt, eds., University of Twente, Netherlands, 1996, pp. 161–172.

[18] C. Lu, J. Hong and S. Cruz-Lara, Emotion detection in textual information by semantic role labeling and web mining techniques. National ChiNan University and Universities of Nancy, 2005.

[19] J. McKay, Generation of idiom-based witticisms to aid second language learning, In: Stock et al., 2002, 77–87.

[20] W. Minker, R. López-Cózar and M. McTear, The role of spoken language dialogue interaction in intelligent environments, Journal of Ambient Intelligence and Smart Environments 1 (2009), pp. 31–36.

[21] J. Morkes, H.K. Kernal and C. Nass, Effects of humor in task-oriented human-computer interaction and computer-mediated communication: A direct test of srct theory, Human-Computer Interaction, 14(4), 1999, pp. 395–435.

[22] A. Nakamura, *Kanjo hyogen jiten* [Dictionary of Emotive Expressions] (in Japanese). Tokyodo Publishing, Tokyo, Japan, 1993.

[23] J.C. Overholser, Sense of humor when coping with life stress, Personality & Individual Differences, 13(7), 1992, pp. 799–804.

[24] M. Ptaszynski, Boisterous language. Analysis of structures and semiotic functions of emotive expressions in conversation on Japanese Internet bulletin board forum – 2channel (in Japanese), M.A. Dissertation, UAM, Poznan, Poland, 2006.

[25] M. Ptaszynski, P. Dybala, S. Higuchi, R. Rzepka and K. Araki, Affect as Information about Users' Attitudes to Conversational Agents, Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'08), Second International Workshop on Human Aspects in Ambient Intelligence (HAI'08), Sydney, Australia, 2008, pp. 459–500.

[26] M. Ptaszynski, P. Dybala, S. Higuchi, R. Rzepka and K. Araki, How to find love in the Internet? Applying Web mining to affect recognition from textual input, Proceedings of the 2008 Empirical Methods for Asian Languages Processing Workshop (EMALP'08) at The Tenth Pacific Rim International Conference on Artificial Intelligence (PRICAI'08), Hanoi, Vietnam, 2008, pp. 67–79.

[27] M. Ptaszynski, P. Dybala, Shi Wenhan, R. Rzepka and K. Araki, A System for Affect Analysis of Utterances in Japanese Supported with Web Mining, Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, Special Issue on Kansei Retrieval, Vol. 21, No. 2 (April 2009), pp. 30–49 (194–213).

[28] G. Ritchie, R. Manurung, H. Pain, A. Waller, R. Black and D. O'Mara, A practical application of computational humour, Proceedings of the 4th International Joint Conference on Computational Creativity, 2007, pp. 91–98.

[29] J.A. Russell, A circumplex model of affect, Journal of Personality and Social Psychology, 39(6), 1980, pp. 1161–1178.

[30] R. Rzepka, Ge Yali and K. Araki, Common Sense from the Web? Naturalness of Everyday Knowledge Retrieved from WWW, Journal of Advanced Computational Intelligence and Intelligent Informatics, 10(6), 2006, pp. 868–875.

[31] N. Schwarz, and G.L. Clore, Mood, misattribution, and judgments of well-being: Informative and directive functions of affective states, J Pers Soc Psychol, 45, 1983, pp. 513–523.

[32] S. Svebak, Revised questionnaire on the sense of humor, Scandinavian Journal of Psychology, 15(2), 1974, pp. 99–107.

[33] A. Szabo, The acute effect of humor and exercise on mood and anxiety, Journal of Leisure Research, 35(2), 2003, pp. 152–162.

[34] M. Takahashi, *Web ni yoru kyoukihindo to n-gram moderu wo mochiita hatsuwabunnseiseishuhou* (Utterance Generation Method Using Web Search Results and Word n-grams), Bachelor dissertation, Hokkaido University, Sapporo, Japan, 2009.

[35] O. Takizawa, M. Yanagida, A. Ito and H. Isahara, On computational processing of rhetorical expressions – puns, ironies and tautologies, Proceedings of The International Workshop on Computational Humor, Netherlands, 1996, pp. 39–52.

[36] H.W. Tinholt and A. Nijholt, Computational Humour: Utilizing Cross-Reference Ambiguity for Conversational Jokes. Proceedings of 7th International Workshop on Fuzzy Logic and Applications (WILF 2007), Camogli (Genova), Italy. LNAI Vol. 4578 (2007). Springer Verlag, pp. 477–483.

[37] A.P. Vilaythong, R.C. Arnau, D.H. Rosen and N. Mascard, Humor and hope: Can humor increase hope?, Humor: International Journal of Humor Research 16(1), 2003, pp. 79–89.

[38] Shi Wenhan, R. Rzepka and K. Araki, User Textual Input Using Causal Associations from the Internet, FIT2008, pp. 267–268.

[39] Y. Yamashita, Kara, Node, Te-Conjunctions which express cause or reason in Japanese (in Japanese), Journal of the International Student Center, 3, Hokkaido University, 1999.