Daboo: An interactive system to make a user guess a word as fast as possible (without using taboo words)

Stefan Marti and Keith Emnett
The Media Laboratory
Massachusetts Institute of Technology
20 Ames Street, Cambridge, MA USA 02139
stefanm@media.mit.edu, emnett@media.mit.edu

Abstract

This paper presents an architecture and a real time computer system for the automatic generation of text in the specific context of the word guessing game Taboo™. To achieve the game's goal – let the user guess a word as fast as possible without using certain taboo words – our system Daboo uses sophisticated algorithms to model user knowledge and to interpret semantically the user input. The former is very important to gradually enhance the performance of the system by adapting to the user's strongest "context" of knowledge. The latter helps bridge the gap between a guess and the actual word to guess by creating a semantic relationship between the two. For this purpose we rely on the semantic inheritance tree of WordNet. Daboo acts effectively as the clue giving party of a Taboo™ session by interactively generating textual descriptions in real time.

1 Introduction

In the process of producing discourse, speakers and writers must decide what they want to say and how to present it effectively. A lot of work has been done in automatic text generation: e.g., McKeown (1985) tried to identify and formalize principles of discourse so that they could be used in a computational process, and she developed TEXT, a system which generates paragraph-length responses to questions about database structure. Moser and Moore (1995) looked at it in the context of better text generation for the explanation component of a training system for technicians. However, none of them looked at these issues in the context of a word guessing game.

A part of automatic text generation (and of specific interest for our purpose), is how to define things verbally. In this context, the work of Milosavljevic and Dale (1996) has to be mentioned. They created PEBA-II, a system which generates encyclopedia descriptions of entities, based on an underlying taxonomic knowledge base. It uses two high level discourse plans: identify and compare-and-contrast. The identify discourse plan is used to describe an entity and the compare-and-contrast discourse plan is used to compare two entities. The current descriptions are limited to the animal domain.

Someone who did a lot of research on a subject we'll also cover was Rosch (1978). She did a series of experiments which showed how concepts appeared to result in a *basic level of categorization*, as opposed to a *superordinate* or *subordinate level*, e.g.,

- *furniture* = superordinate
- *chair* = basic
- *kitchen chair* = subordinate

She found that the basic level of categorization is the level most quickly recognized as a perceptual gestalt by adults, and most quickly learned by young children. It frequently represents a level where a whole is articulated into functionally specific parts. It is not the lowest most specific level, nor the most abstract, but always somewhere in the middle of the relationship hierarchy.

2 Taboo™ – the real game

In the past six years, this game has received more play with a wider group of people than any other game in the party games domain. But its success is not limited to the English version: An independent market research institute surveyed 1000 assorted game outlets in Germany to determine the "game hits of 1996", which were the ten games with the strongest turnovers in Germany. Taboo™ has been in first place since 1994.

Following Sarrett (1993), Taboo™ is played by two teams of at least two players each. The object of the game is to correctly identify as many words as possible in one minute. Taboo™ comes with a deck of double-sided word cards. Each side contains six words – one in large, colored print at the top, the rest in smaller black print beneath. On a team's turn, one member of the team becomes the clue giver and loads a stack of cards into the provided card holder. When the timer is started, the clue giver tries to get his/her teammates to say the word at the top of the first card. S/he can't use gestures or "sounds like" clues but must accomplish the task through verbal clues only. But here's the catch: the clue giver is forbidden to say any form of any word on the card. For example, a card might have *camel* as the colored word and *animal, desert, hump, spit*, and *cigarettes* as the taboo words. The clue giver would have to find a way to describe *camel* without using any of those words.

Although this sounds like a simple task, it is not necessarily. There are usually plenty of ways to succeed. The problem is, once you look at the taboo words you tend to get stuck on them and it's difficult to knock your thought processes onto a different track. It's kind of like trying not to think of a pink elephant. And in Taboo™, the other team is looking over your shoulder to make sure that you don't screw up. If you use one of the taboo words, they'll sound the game's buzzer and you'll forfeit a point and valuable time. You can pass if you're stumped, but it costs you a point also. When a word is guessed, buzzed, or passed, you move on to the next card. When time expires, your team earns one point for every word guessed (and loses one for each passed or buzzed word). Then the other team gets a shot at it. It is usually played until everyone on every team has had a chance to give clues.

Team members are free to guess at the word as many times as needed until they get it right. Although theoretically there can be any number of people on a team, volume makes it difficult to distinguish answers once teams reach eight or so.

3 Daboo – the system

Our goal was to implement the clue-giving party on a computer system. In order to realize that, we had to restrict the original game in various ways:

- We do not use speech input or output. Daboo is text based. The user types in guesses in a text window, and Daboo prints out clues and descriptions in another window.
- Daboo only uses nouns: all the words to guess are nouns, and therefore the user has to limit guesses to nouns. This was a restriction to avoid ungrammatical outputs generated from our semantic inheritance tree.

- There is no second team in Daboo – it's only the user and the computer. So no other team is looking over your shoulder to make sure that you don't screw up with taboo words, but there is also no reason to assume this would ever happen, since computers are quite reliable that way...
- There is no game buzzer to sound when the clue giver uses a taboo word, because Daboo has built-in security features so that it should *never ever* use taboo words. (Nevertheless it can happen, e.g., if there are hyphens or other special characters in the expressions it gets from the semantic inheritance tree.)
- In Daboo, the user can't skip a word when s/he's stuck. But Daboo does that automatically if it thinks the user will never find out the specific word.
- The original game is about guessing as many words as possible in one minute. Instead of this idea, a Daboo session is about how long it takes a user to guess all the available cards. (Currently there are five cards available: *hamburger, airplane, star, rose, spice*.)

Our system uses a Java™ server to communicate with external PERL scripts and two Java™ clients (applets) that connect across the network via sockets. One applet is for user input and the other is for clue output. The system simulates the typing of an actual person by adding a delay between each letter that is sent to the output window. You can see an example of the clue applet in Figure 1 below, and an example of the guess applet in Figure 2.

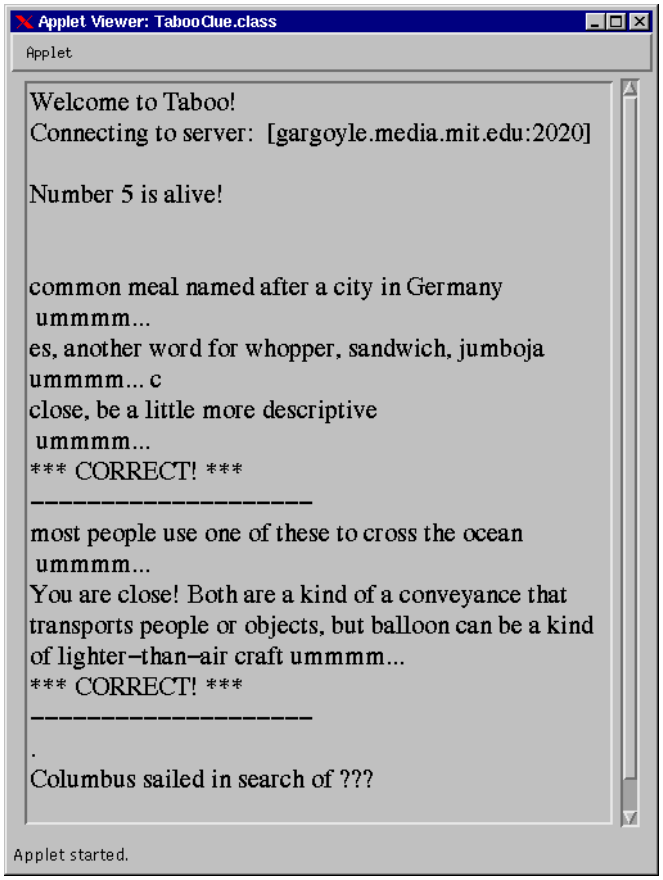


Figure 1: Daboo clue applet window

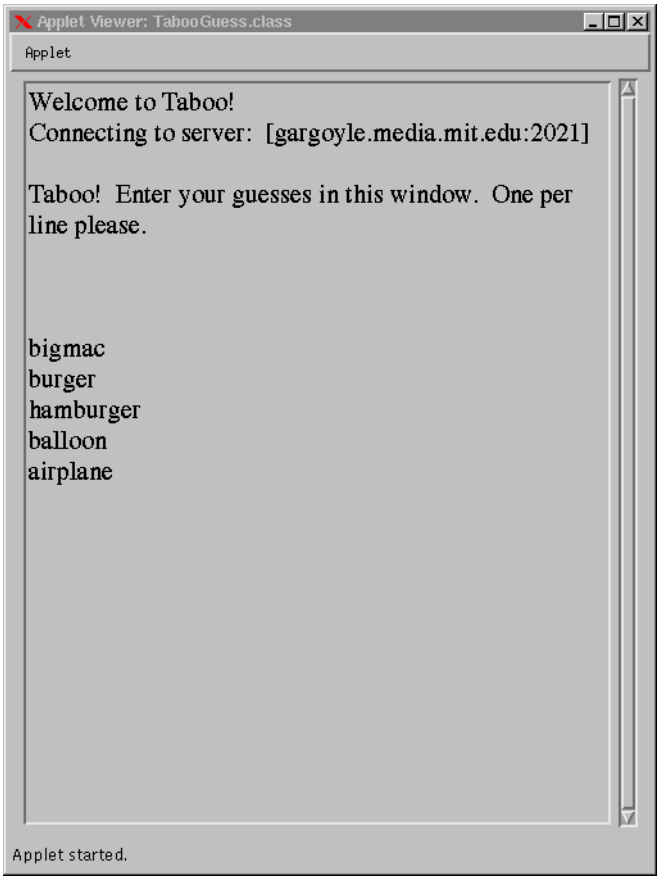


Figure 2: Daboo guess applet window

3.1 Interactivity

Due to the nature of the game Taboo™, a primary goal of our system is to lead the user to the correct answer as fast as possible. The best way to achieve this goal is through the user of an interactive system that intelligently responds to user input. Our system is task-oriented in that it tries to lead the user to a specific, correct answer. We also designed it as a cooperative system, assuming the person playing the game shares this common goal, and will attempt to work with the system to achieve it. With these characteristics in mind, we implemented two major features which facilitate constructive interaction:

- The user can guess a word at any time – s/he does not have to wait for the system to complete an utterance and then prompt the user for a guess.
- The system will immediately respond to any guess the user makes, even if the guess does not change the system's current descriptive method.

The first feature implements an important aspect of turn-taking described by Duncan (1974) – back channel auditor response. We cannot really describe the interaction as a series of turns, because the clue-giver never gives up the speaker role. Nevertheless, the system's ability to accept and respond to input at any given time helps facilitate the speaker-auditor interaction. The system is quite limited in that it cannot respond to visual or auditory cues. It can only respond to typed, single word guesses; however, these guesses still contain a wealth of information. In this sense, the user input is more than a simple back channel response. The second feature stems from the speaker's responsibility to intelligently evaluate the user's input, and even more simply, to indicate to the user that the system has received and understands the user's input. This principle of responding to each piece of information provided by the user was described as a design maxim by Grice (1975). Our system uses several short phrases to describe how the system has reacted to the user's input. The computer outputs "*ummmm...*" immediately after accepting an input from the user. This serves as a temporary acknowledgement until the computer has completed its series of evaluation algorithms. We want to describe this as a speaker back-channel response, because the computer still maintains control of the floor. If the guess causes a change in descriptive strategy, then the computer will output a phrase that describes how close the guess was, which helps implement a smooth transition from one strategy to the next. A phrase such as "*You are very close...*" reaffirms the user's train of thought, whereas "*Think of this...*" implies that the user is not very close, and that the system is going to try another strategy. If the system cannot determine any meaning for the guess, it types "*(no)*" as an aside and continues with the current strategy. In the following example, the word is *spice* and the user inputs *car* as a guess:

> *cinnamon and oregano are each*
> *ummmm... (no) a kind of spice*

The Taboo™ game itself as well as Daboo can be regarded as an untypical but nevertheless (on mutual agreement) strongly institutionalized form of ellipsis. Therefore it is not astonishing that Allen (1995) proposes a "semantic closeness check" between the input fragment of a sentence and the potential target fragments to deal with ellipsis. We will describe a possible implementation of this idea later in this paper (section 3.3.4).

3.2 User knowledge modeling

The basic concept that the game Taboo™ explores is how different people describe the words on the cards, and why they choose the particular methods they do. In fact, you will

see that much of the discussion between turns centers around explaining the thought processes that the participants went through (Grantham 1997). We believe the descriptions people use draw not only on the knowledge they possess, but also on the knowledge they believe their team members possess. This belief is supported by the work of Don (1990) in computer narrative. She describes the human/computer interface in terms of the relationship between generated narrative and the knowledge and experience of the person interacting with the computer. Both the content and meaning of the discourse are affected by the context. In other words, the user will only understand what the clue-giver is trying to describe if they share a common knowledge about that topic. If a person uses the term "Sleepless in" to elicit the word "Seattle," then s/he is assuming (or hoping) the guesser has seen or otherwise is aware of this movie. Levinson describes this common knowledge between a speaker and hearer as a participant role (Levinson 1988), also referred to as a footing (Goffman 1981).

With these ideas in mind, we attempt to model the knowledge of the user and provide descriptions that are relevant to the user's strongest "context" of knowledge. Our system cannot generate knowledge domain specific descriptions on-the-fly (all of our descriptions are preprogrammed), but it can react to user input and select the context it determines is most appropriate. The system characterizes a user's knowledge according to six broad categories, borrowed from the game Trivial Pursuit™:

- Entertainment
- Sports and Leisure
- Science and Nature
- History
- Arts and Literature
- Geography

For every card in the database, we have a corresponding set of descriptions that each fit into one of the categories listed above. For example, the word *star* uses as the entertainment clue: *Luke Skywalker was in ??? Wars*. To evaluate the knowledge of the user, we try to match each guess the user inputs to one of the six knowledge domains. We admit that modeling knowledge based on a single word response is very weak in many instances, but nevertheless we chose it for simplicity. Each of the knowledge contexts has a large database of words and information associated with that topic. Every time the user inputs a guess, the system scans the databases for occurrences of that word, and assigns it to the context (if any) that produces the most hits. The model therefore continuously updates itself as the game progresses.

The basic strategy for providing knowledge specific descriptions of words is:

1. Choose the strongest knowledge context initially, and output the corresponding description of the word to guess.
2. Evaluate whether the user understands the chosen context.
3. If not, switch to the second-strongest context and attempt another description.
4. If this fails, pass and proceed to the next word.

In step two, we use the input of the user to evaluate if s/he has understood the description the system gave. If the user enters two guesses that are not in the same context as the description, or if the user fails to input a guess within a certain period of time, the system assumes the user is not on the "same wavelength" and attempts to describe the concept in a different way, essentially inducing a change in footing. This sequence can also be interrupted by several other algorithms (described later) that attempt to determine if the user is close to the correct answer.

3.3 Semantic interpretation of user input

Each word the user types in is analyzed very carefully by Daboo. This is necessary to get some semantic knowledge out of the simple strings of characters the user types in.

Daboo has different algorithms to find out more about a word. The simple ones give back the confirmation that the word has a certain characteristic (e.g., being a synonym to the word to guess). A more complicated one gives back complete sentences (which are generated on the fly and therefore take a lot of computer power).

In the current version of Daboo, there are four algorithms implemented for semantic interpretation of user input:

- guess is part of word to guess
- guess is a taboo word
- guess is synonymous to the word to guess
- determining the semantic relationship between guess and word to guess

3.3.1. Guess is part of word to guess

Typing in a word (e.g., *burger*) that is part of the word to guess (*hamburger*) is not a successful guess, but a very close one. Therefore we decided to treat these user guesses separately. Such a word falls in the semantic category "close guess" and Daboo spits out the comment:

Close, be a little more descriptive!

Forcing the user to be more descriptive is a consequence of the idea that such a guess is very close because it is either a word stem or the word to guess is a concatenated word:

- user guess "meter" misses the *affix* "kilo" for word to guess "kilometer"
- user guess "interesting" misses the *prefix* "un-" for word to guess "uninteresting"
- user guess "drink" misses the *suffix* "-able" for word to guess "drinkable"
- user guess "sun" is part of the *concatenated* word to guess "sunrise"

3.3.2. Guess is a taboo word

As the user does not know which words are taboo for Daboo, it is quite probable that s/he will happen to type in a taboo word at some point. As Daboo must not use these words, it cannot repeat the guess, but is allowed to mention that it is a word which it can't use to describe the word to guess. Daboo comments a user input which is a taboo word by the following sentence:

Aaah, almost! That word is a taboo word!

This might be important information for the user as taboo words are usually words which are within the first associations of the word which is to guess. The more taboo words the user knows the faster s/he will guess the word. By knowing two taboo words, an experienced Daboo player might guess the word immediately.

3.3.3. Guess is synonymous to the word to guess

Another important clue for the user is if his/her guess is a synonym to the word to guess. This is obvious, as synonymy means "similarity of meaning". But as Miller et al. (1993) mentioned, there are a lot of problems with the practical use of the abstract term synonym. According to the common definition, two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made. By that definition, true synonyms are rare, if they exist at all. So a weakened version restricts truth of value to a certain linguistic context. For example, the substitution of *thrower* for *pitcher* does not change truth value in a sports context, although in a kitchen context this substitution is nonsense.

But in the context of Daboo, this isn't a real problem: If the word to guess is *pitcher*, we are willing to accept synonyms of *every possible* context, including kitchen, to make the user find the word as fast as possible. As

the word to guess is a single word without a specific context, it just doesn't matter if the words would change the truth value in a specific context.

But our semantic inheritance tree WordNet (see Miller et al. 1993, Hiyakumoto et al. 1997) does care about different contexts (senses), and therefore our first idea to let WordNet search for synonyms turned out to be not a very realistic one. We had to accept the fact that most synonyms WordNet creates are more like descriptions than single word synonyms. Although descriptions can be also very interesting for Daboo (and we do use them in a different algorithm extensively), we decided to create our own synonym lists. Our decision was supported by the fact that synonymy is undirected (*bench* is synonymous to *bank,* as well as *bank* is synonymous to *bench*), and that there was no reason to generate a synonym list of the word to guess on the fly.

The Daboo algorithm to check synonymy between guess (*bigmac*) and word to guess (*hamburger*) looks up a preset list of single word synonyms (*bigmac, whopper, sandwich, jumbojack*). If it finds the user's guess within this list, it generates a sentence which contains all synonyms except the user's guess:

> *Yes, another word for whopper, sandwich, jumbojack!*

The quality of information the user gets by this feedback depends strongly on the accuracy of the synonyms. To generate this list is mainly a question of creativity and heavy domain knowledge, which, in the case of *hamburger*, is also culture specific. This means not only western as opposed to other cultures, but also, as the *hamburger* example might show, down to regional characteristics.

3.3.4. Determining the semantic relationship between the guess and the word to guess

Another algorithm which should create an utterance to help the user guess the word is based on a commonly seen strategy of clue-giving used by people playing the real Taboo™ game: What does the word the partner just uttered have in common with the word to guess? How could I bridge the gap between them? How could I describe the relation between the two words? Or in a more linguistic style: Try to define the semantic relationship between the word to guess and the user's guess!

Milosavljevic and Dale (1996) say it clearly: If we want to get computers to generate descriptions of objects the way we do, we need to build mechanisms for generating comparisons between entities. Obviously, this is a quite demanding task for an auto-

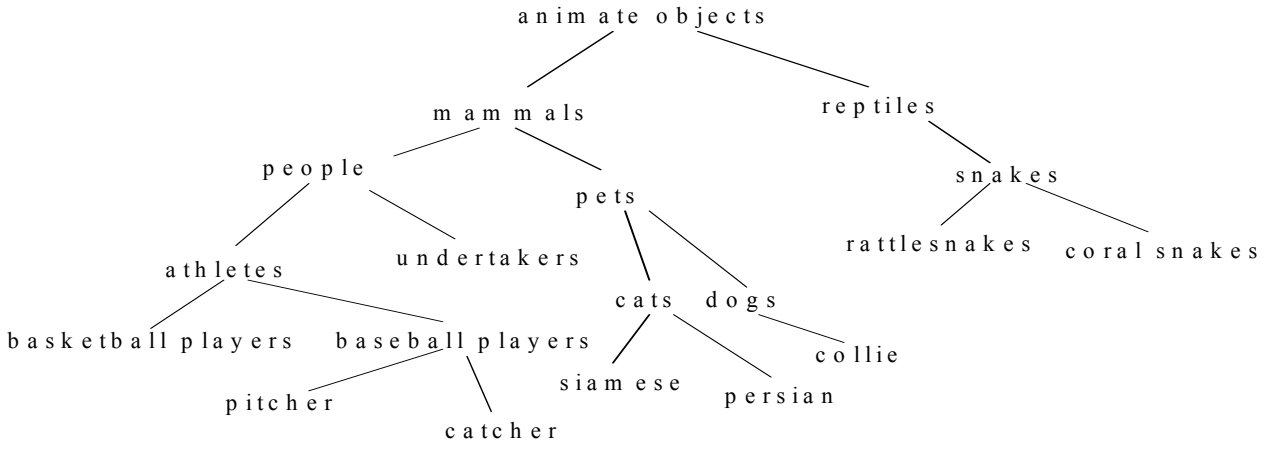


Figure 3: Pseudo inheritance tree

matic clue giving system like Daboo. First, this requires additional knowledge besides the preset lists of descriptions and synonyms. The system has to "understand" the user's guess in a general semantic way to make a connection to the word to guess. Second, it is a highly interactive task, because each combination of guess and word to guess do have a very specific semantic relation.

To make a connection of this kind we need a semantic tree or inheritance tree, which describes the "a kind of" relation of all possible words. An example of such a tree is illustrated in Figure 3 (similar to Schmandt 1994). An example for the semantic relation between the words *siamese* and *snake* is shown in Figure 4. To determine their relationship, one has to find the closest path that connects the nodes of the tree. Usually, one has to travel up the specific branch or limb of each word until it meets with the branch of the other word. The node where they meet is the *most common superordinate*. This is the characteristic that both words have in common. In our example, it is *animate object*: both *snake* and *siamese* are a kind of *animate object*. As the user knows only his/her guess (*snake*), mentioning the most common semantic property of both words (*animate object*) is interesting, but not very helpful, because there are a lot of animate objects beside snakes. So the user would be glad to get another hint, something like a "signpost" – where to go in the tree from the most common superordinate. The nodes right beneath it have this function: they explain the first indication of why the two words are different. We call these two the *branching description*: it describes how the tree for both words branches at the most common superordinate. In our example, both words are a kind of *animate object*, but *snake* is a kind of *reptile*, and *siamese* is a kind of *mammal*.
And that's almost the sentence the algorithm creates when it is fed with the words *snake* and *reptile*. As *siamese* is the word to guess, it is obvious that Daboo can't mention it, so the basic content of the sentence would be:

> *Both are a kind of animate object, but snake is a kind of reptile, and ??? is a kind of mammal.*

The sentence describes the common aspect of the two words, and also what makes them different.

As mentioned before, this is not an unequivocal method of defining a word, it is just a hint. But as each hint can be regarded

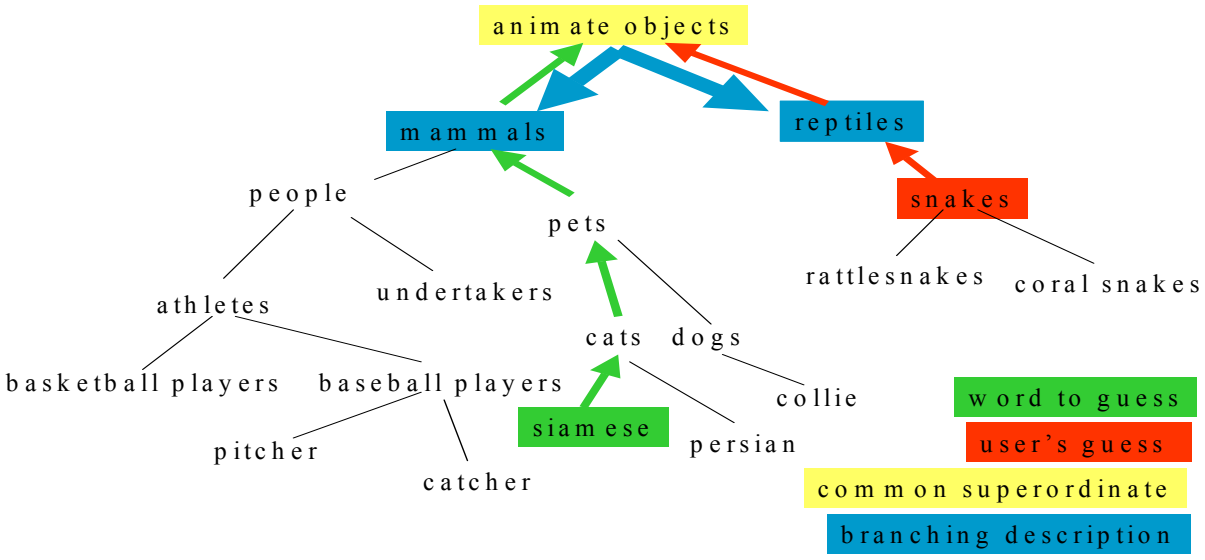


Figure 4: Semantic relation between *siamese* and *snake*

as a vector in a two dimensional space (e.g., starting from *animate object* and pointing towards *mammals*), two vectors would cross in a single point. Having more than one hint of this sort (from two or more guesses) should enable the user to locate "spatially" the word to guess in the semantic tree. We would like to describe this as a sort of "semantic triangulation process". As the semantic tree is not a real two dimensional object, but a mere description of relations, this crossing of vectors is obviously only symbolic, but nevertheless very helpful!

To realize such an algorithm we rely heavily on the hypernym function of WordNet. For a single word, the hypernym feature of WordNet gives the upper part of the semantic tree, starting from the word itself (e.g., *snake*) and ending usually at the top of tree (*entity*).
So basically one has to compare the branches of the guess and the word to guess, and sooner or later they will meet. This would be a quite easy task, but as mentioned before, WordNet very often gives out different senses for a word (see Figure 5), which makes comparison a complex and computationally costly process. Each sense of a word is a node which can be located virtually anywhere on the semantic tree. (The version of WordNet used for Daboo consists of 168,135 nodes, each containing several words and a description.)

To find the connection for two words, we actually have to compare several branches of one word with several branches of the other word. So the algorithm not only has to find the connection between two words (say: most common superordinate), but the *closest* of *a lot of* common superordinates. (We define closeness as simply how many nodes exist in between the two words. The actual value we call the *Nearness Level*). This is a very demanding problem computationally, similar to finding the closest street connection between two addresses on a city map. Actually, what the algorithm tries to find is

```
Synonyms/Hypernyms (Ordered by Frequency) of noun snake
4 senses of snake

Sense 1
snake, serpent, ophidian -- (limbless scaly elongate reptile; some are venomous)
       => diapsid, diapsid reptile -- (reptile having a pair of openings in the skull behind each eye)
           => reptile, reptilian -- (any cold-blooded vertebrate of the class Reptilia including tortoises turtles
              snakes lizards alligators crocodiles and extinct forms)
               => vertebrate, craniate -- (animals having a bony or cartilagenous skeleton with a segmented spinal
                  column and a large brain enclosed in a skull or cranium)
                   => chordate -- (animal having a notochord)
                       => animal, animate being, beast, brute, creature, fauna -- (a living organism characterized by voluntary movement)
                           => life form, organism, being, living thing -- (any living entity)
                               => entity -- (something having concrete existence; living or nonliving)

Sense 2
snake, snake in the grass -- (a deceitful or treacherous person)
       => bad person -- (a person who does harm to others)
           => person, individual, someone, mortal, human, soul -- (a human being; "there was too much for one person to do")
               => life form, organism, being, living thing -- (any living entity)
                   => entity -- (something having concrete existence; living or nonliving)
                      => causal agent, cause, causal agency -- (any entity that causes events to happen)
                         => entity -- (something having concrete existence; living or nonliving)

Sense 3
Snake, Snake River -- (a tributary of the Columbia River)
       => river -- (a large stream of water)
           => stream, watercourse -- (a natural body of running water flowing on or under the earth)
               => body of water, water -- (the part of the earth's surface covered with water)
                   => object, inanimate object, physical object -- (a nonliving entity)
                       => entity -- (something having concrete existence; living or nonliving)

Sense 4
snake, plumber's snake, auger -- (a long flexible steel coil for dislodging stoppages in curved pipes)
       => hand tool -- (a tool used with workers' hands)
           => tool -- (an implement used in the practice of a vocation)
               => implement -- (a piece of equipment or tool used to effect an end)
                   => instrumentality, instrumentation -- (an artifact (or system of artifacts) that is instrumental in
                      accomplishing some end)
                       => artifact, artefact -- (a man-made object)
                           => object, inanimate object, physical object -- (a nonliving entity)
                               => entity -- (something having concrete existence; living or nonliving)
```

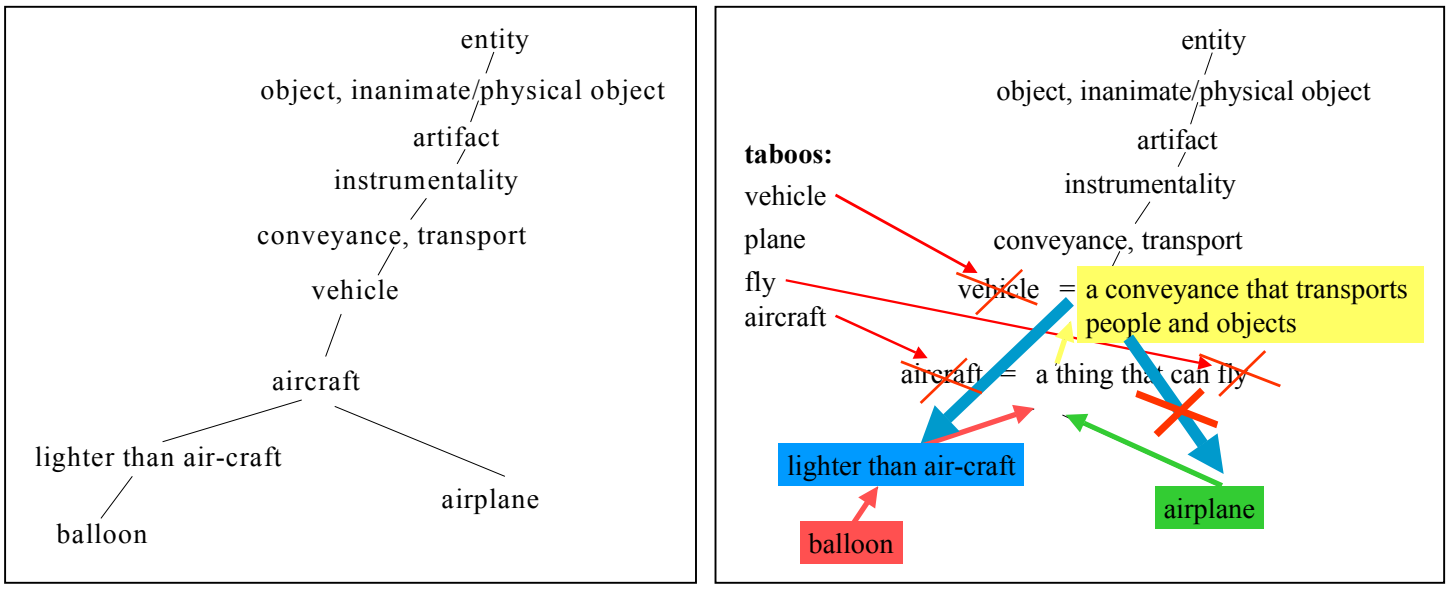
Figure 5: Original WordNet hypernym branches for the 4 senses of *snake*

the shortest connection of *several* houses (each sense of a word) to *several other* houses (each senses of the other word) on a map. Describing the actual search algorithm wouldn't be appropriate for this paper, but at least one has to understand that there is no guarantee that a connection it finds is the optimal one.

Although the algorithm is already complex, it gets even more complex because Daboo is not allowed to use certain taboo words for its descriptions. E.g., if the *closest common superordinate* is a taboo word, it has to be replaced. For this purpose we use the short descriptions WordNet provides for each word instead of real one word synonyms. But this is a truly recursive process: Within the descriptions for taboo words, again all the taboo words have to be replaced! Although this sounds like really bad luck, it happens quite often, because many taboo words are semantically close to the word to guess. As it is sometimes not possible to replace a taboo word by its definition (because there is none, or for other reasons), Daboo eventually replaces such words by "XXX". And again, it sounds like really bad luck, but conforming to Murphy's Law, quite often there are multiple "XXX" replacements, which make the generated sentence almost unreadable.

Another reason that a generated sentence may not be very helpful is if the path to connect the guess and the word to guess is just too far, meaning too many nodes are in between, or the *closest common superordinate* is something like *entity*. It is not very helpful to know that both the guess and the word to guess are an *entity*, but the guess (e.g., *hamburger*), is an *inanimate object*, and the word to guess an *animate object*. (Although this utterance is of course true and could be helpful, the user just seems to be too far away from the word to guess, and for such situations Daboo has more efficient strategies than this one.)

The generated sentence shown above can be further modified in two ways, depending on the computed *Nearness Level*.
Although it is good if the two words are close (a low *Nearness Level* means their branches meet quite soon), it is bad if they are *too* close: quite often the branching description is the word itself. For example,



Original tree *Tree after replacements*

Figure 6: Example for hypernym algorithm processing

guess is *hamburger*, word to guess *hotdog*. The closest common superordinate of *hamburger* and *hotdog* in WordNet is *sandwich*, which is just one node away from both! So the output sentence has to be adjusted, mainly by skipping certain parts. In our example, the following utterance is generated:

Both are a kind of sandwich.

Depending on whether the *first,* the *second,* or *both* words are too close to the branching descriptions, we choose between three structurally different templates of the sentence describing the semantic relationship.

And finally, a last modification is made to help the user guess the word. Depending on the *Nearness Level,* an additional utterance is put in front of the already generated sentence. It ranges from *You are very close* to *Not quite*!

So the output for the guess *balloon* and the word to guess *airplane* is:

You are close! Both are a kind of a conveyance that transports people and objects, but a balloon is a kind of lighter than air-craft.

In this example (see Figure 6), the closest common superordinate of *balloon* and *airplane* would be *aircraft*. As this is a taboo word, it is replaced by the expression *a thing that can fly*. But as *fly* is also a taboo word, the next definition for *airplane* is extracted, which is *vehicle.* This again is a taboo, and only the third try is successful, which is *a conveyance that transports people and objects*. As the branching description from *aircraft* to *airplane* is the word *airplane* itself (meaning the closest common superordinate of *balloon* and *aircraft* is just one node above the word *airplane*), it is left out when the sentence is created. (Note: The expression *air-craft* in the branching description for *balloon* should be replaced because *aircraft* is a taboo, but was not matched because of the hyphen.) As the *Nearness Level* is quite low (2), the expression *You are close!* is added in front of the whole sentence.

4 Future Work

The Daboo system was limited to a large extent by its text only mode of communication. We did not have access to common conversational cues such as intonation and facial expressions, which are still permitted by the rules of the game. The first step here would be a speech interface using a voice processor for the user's guesses and text-to-speech synthesis for the computer output. An even more complex interface could use digital video processing to extract facial expressions from the user and an animated avatar to express the ideas of the clue-giver. These interfaces are considerably more complex, and the effects they may have on the interaction is still an ongoing area of research.

Our knowledge model focused only on general areas of knowledge. This model could be extended to include information about age, interests, background, geographical location, culture, and many other features that shape the way a person thinks and the knowledge they possess. All of these characteristics are potentially taken into account by the clue-giver when determining the best way to describe a word to a particular audience.

5 Results and Conclusions

Our attempt to model the knowledge of the user based on one-word responses met with limited success. First of all, our descriptions were generally common enough that most people could guess the correct answer from any of them. Second, we really want to model concepts and ideas, which require more than a single word to express. If we

could match phrases to a more comprehensive knowledge database, we think we could achieve much better results. We do feel, however, that the six categories we chose to represent knowledge contain about the right amount of breadth and separation to pinpoint an area a user could be most comfortable with.

The use of our response strategies that attempted to illustrate the relationship between the user's guess and the target word proved somewhat difficult to evaluate. They many times provided accurate and useful descriptions of the similarities and differences between the two words; but, because the original descriptions were so simple and intuitively obvious, we found it hard to judge the effectiveness of the response strategies. With more difficult words, we think these strategies could be more useful, although the hypernym descriptions generated were sometimes too long for the fast paced interaction that needs to occur.

References

Allen, James. 1995. *Natural Language Understanding* (Second Edition). Benjamin Cummings.

Don, Abbe. 1990. Narrative and the interface, in Brenda Laurel (ed.): *The Art of Human-Computer Interface Design*. Reading: Addison-Wesley.

Duncan, S., Jr. 1974. *On the Structure of Speaker-Auditor Interaction During Speaking Turns*. Language in Society, 2:161-180.

Grantham, Sola. 1997. Transcription of a Taboo™ session. Unpublished data. Cambridge: Media Lab, Massachusetts Institute of Technology.

Goffman, Erving. 1981. *Forms of Talk*. Philadelphia: University of Pennsylvania Press.

Grice, H.P. 1975. Logic and Conversation, in Cole and Morgan (eds.): *Syntax and Semantics: Speech acts*. Volume 3, pages 41-58, Academic Press.

Hiyakumoto, L., Prevost, S., and Cassell, J. 1997. *Semantic and Discourse Information for Text-to-Speech Intonation*. Massachusetts Institute of Technology.

Levinson, Stephen C. 1988. *Putting linguistics on a proper footing: Explorations in Goffman's concepts of participation*. In: Paul Drew & Anthony Wootton: Erving Goffman: Exploring the Interaction Order. Oxford: Polity Press.

McKeown, Kathleen. 1985. *Text Generation – Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. 1993. *Introduction to WordNet: An On-line Lexical Database.* Technical report, Cognitive Science Laboratory, Princeton University.

Milosavljevic, M., and Dale, R. 1996. *Strategies for Comparison in Encyclopedia Descriptions*, in Proceedings of the Eighth International Natural Language Generation Workshop, Herstmonceux Castle, Sussex, UK, 12-15 June 1996. p. 161-170.

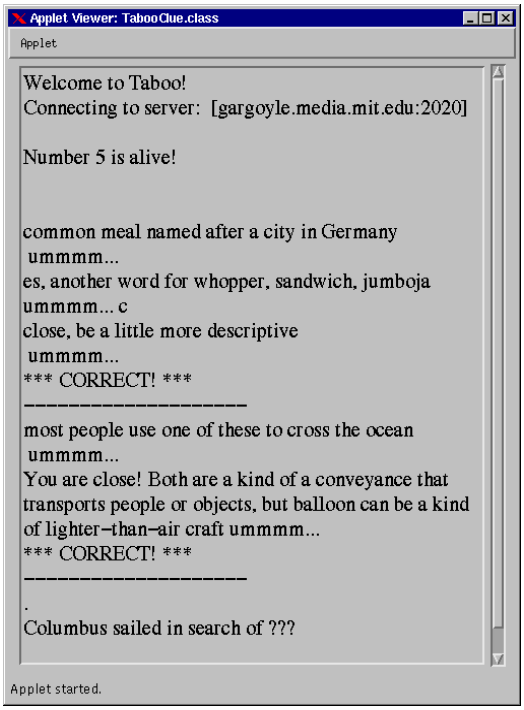
Moser, M.G., and Moore, J.D. 1995. *Investigating Cue Selection and Placement in Tutorial Discourse*, in Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, 130-135, June 1995.

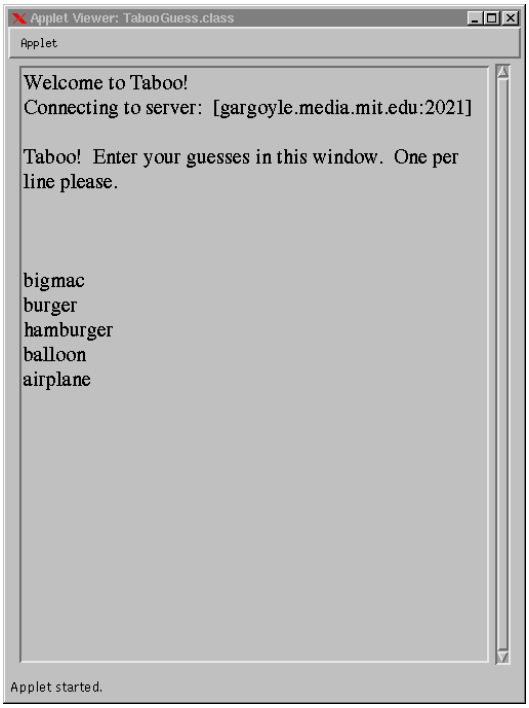
Rosch, Eleanor. 1978. *Principles of categorization*. In Cognition and categorization, ed. by E. Rosch and B. Lloyd. Hillsdale, NJ: Lawrence Erlbaum, pp. 27-48.

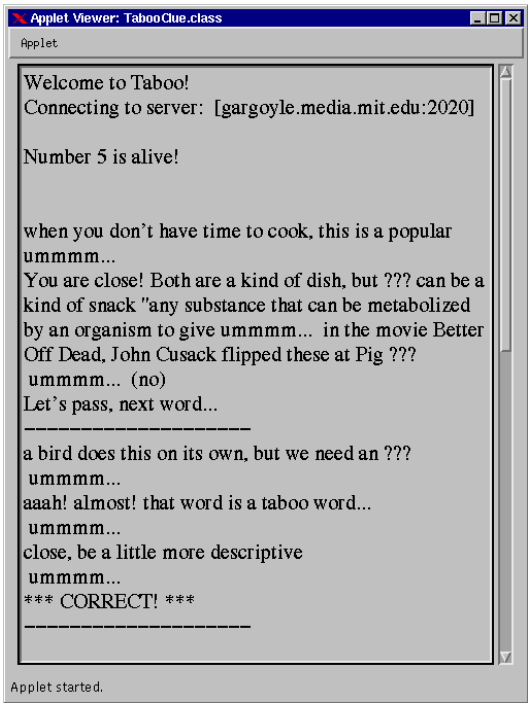
Sarrett, Peter. 1993. Taboo™. The Game Report, issue #1.4.

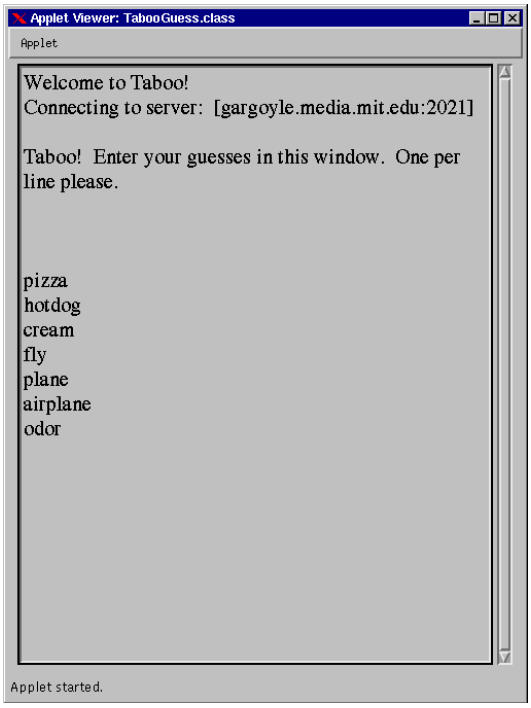
Schmandt, Christopher. 1994. *Voice Communication with Computers*. Van Nostrand Reinhold, New York.

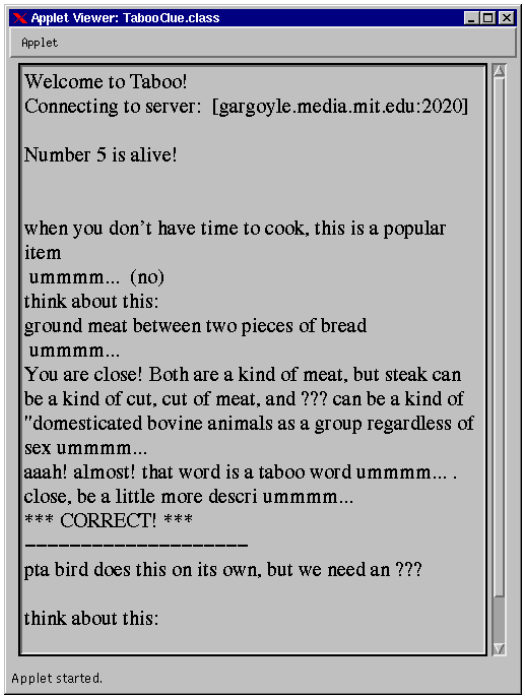
Appendix: session examples

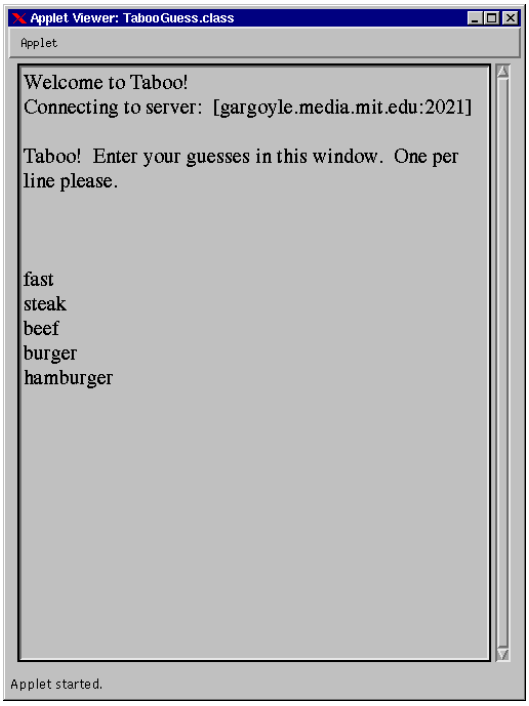


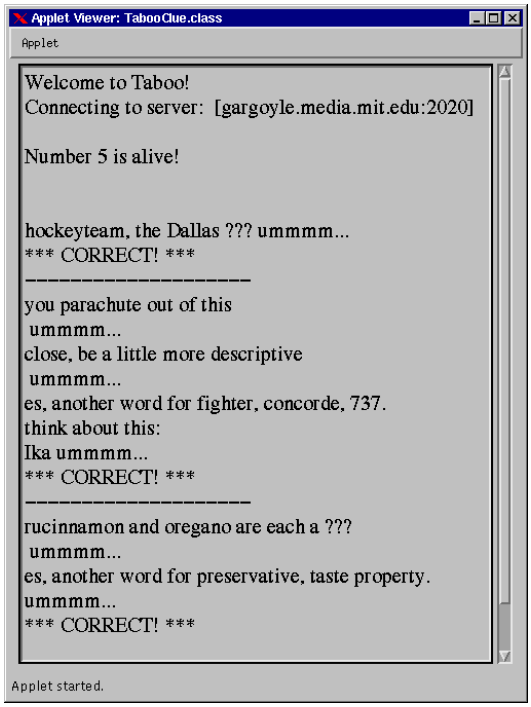


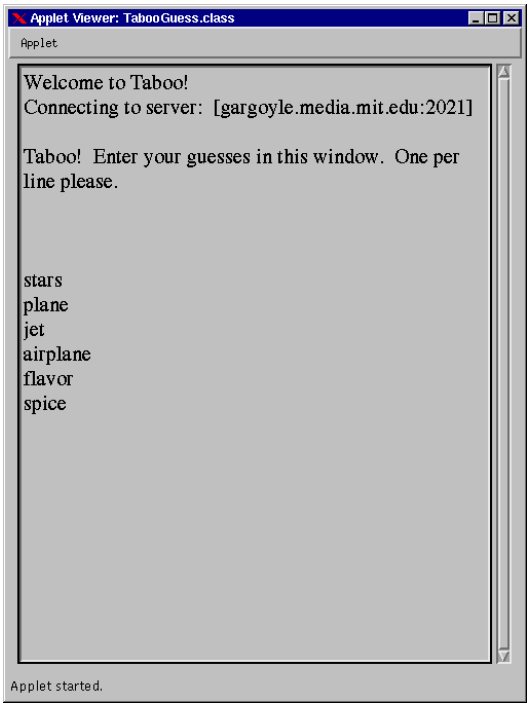





Applet Viewer: TabooClue.class
Applet
Welcome to Taboo!
Connecting to server: [gargoyle.media.mit.edu:2020]
Number 5 is alive!
when you don't have time to cook, this is a popular item
ummmm... (no)
think about this:
ground meat between two pieces of bread
ummmm...
You are close! Both are a kind of meat, but steak can be a kind of cut, cut of meat, and ??? can be a kind of "domesticated bovine animals as a group regardless of sex ummmm...
aaah! almost! that word is a taboo word ummmm... .
close, be a little more descri ummmm...
*** CORRECT! ***
pta bird does this on its own, but we need an ???
think about this:
Applet started.


Applet Viewer: TabooGuess.class
Applet
Welcome to Taboo!
Connecting to server: [gargoyle.media.mit.edu:2021]
Taboo! Enter your guesses in this window. One per line please.
fast
steak
beef
burger
hamburger
Applet started.


Applet Viewer: TabooClue.class
Applet
Welcome to Taboo!
Connecting to server: [gargoyle.media.mit.edu:2020]
Number 5 is alive!
hockeyteam, the Dallas ??? ummmm...
*** CORRECT! ***
you parachute out of this
ummmm...
close, be a little more descriptive
ummmm...
es, another word for fighter, concorde, 737.
think about this:
Ika ummmm...
*** CORRECT! ***
rucinnamon and oregano are each a ???
ummmm...
es, another word for preservative, taste property.
ummmm...
*** CORRECT! ***
Applet started.


Applet Viewer: TabooGuess.class
Applet
Welcome to Taboo!
Connecting to server: [gargoyle.media.mit.edu:2021]
Taboo! Enter your guesses in this window. One per line please.
stars
plane
jet
airplane
flavor
spice
Applet started.