KTH Tangrams Corpus v1.0.2 Overview

Todd Shore
KTH Speech, Music and Hearing
Stockholm, Sweden
tcshore@kth.se

February 1, 2018

Contents

1 Introduction

This document describes the corpus *KTH Tangrams*, which is a collection of task-oriented dialogues situated in an online board game between two human participants. This was done as part of the project COIN — *Co-adaptive human-robot interactive systems* ("Ömsesidig adaption i system för människa-robotinteraktion"), funded by the Swedish Foundation for Strategic Research from July 1, 2016 to June 30, 2021 (reference number RIT15-0133).

Each dialogue in the corpus consists of a pair of participants which alternately assume the roles of **instructor**, who has a visual cue of which piece is to be selected on a shared game board, and **manipulator**, who must select the piece in question with the help of the instructor's commands (see Figure 1). These roles are analogous to those of director and matcher in traditional reference communication tasks, with the terms defined by Schober and Clark [7] but the task itself originating from Krauss and Weinheimer [2]. For further information on the experimental setup, see Todd Shore, Theofronia Androulakaki, and Gabriel Skantze. "KTH Tangrams: A Dataset for Research on Alignment and

Conceptual Pacts in Task-Oriented Dialogue." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. (Miyazaki, Japan, May 7–12, 2018). Ed. by Nicoletta Calzolari et al. To appear. Paris, France: European Language Resources Association (ELRA).

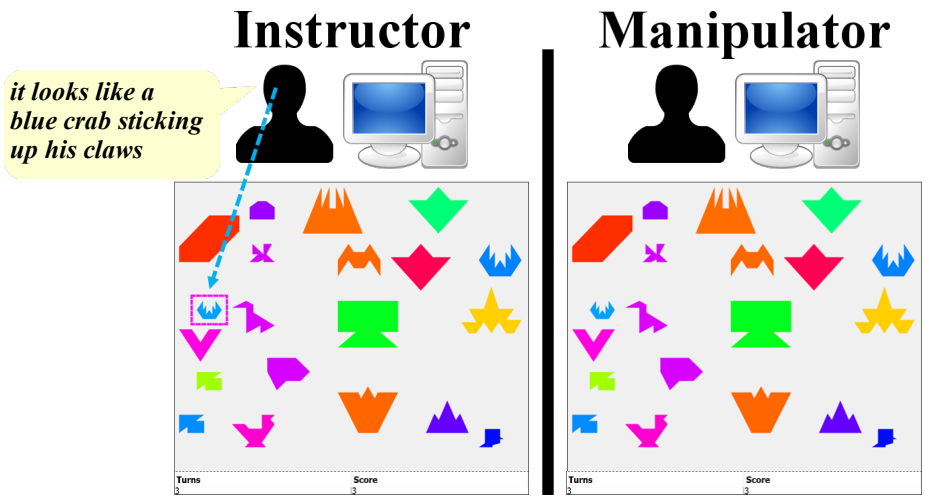

Figure 1: The game board as seen by the respective roles.

2 Version

This document is intended for KTH Tangrams v1.0.2. The versioning scheme follows `MAJOR.MINOR.PATCH`:

`MAJOR` denotes changes to the dataset structure and/or annotation scheme.

`MINOR` denotes changes in the amount of data present, e.g. by adding more dialogues.

`PATCH` denotes changes in the transcriptions/annotations for previously-existing dialogues, e.g. fixing transcription errors.

3 Data Structure

The directory `Data` contains the data collected for each dyad, whereby the data for a given dyad is stored in a directory with the dyad ID as the directory name:

```
<DYAD_ID>
└─ screnshots
   └─ round-<ROUND_ID>-<TIMESTAMP>-<PARTICIPANT_ID>.png
```

```
    └── selection-entity<ENTITY_ID>-<TIMESTAMP>-<PARTICIPANT_ID>.png
├── desc.properties
├── events.tsv
├── events-<PARTICIPANT_ID>.txt
├── img-info.tsv
├── participant-metadata.tsv
├── session-metadata.tsv
├── system-<PARTICIPANT_ID>.log
├── utts.tsv
└── utts.xml
```

3.1 Session Metadata

The file `session-metadata.tsv` contains general information about the experimental setup and statistics about the recording session.

`END_SCORE` The participants' score at the time the recording session ended.

`ENTITY_COUNT` The number of unique entities in the game.

`EVENT_COUNT` The number of events which occurred in the game.

`EXPERIMENT_VERSION` A description of the exact version of experiment software used, comprised of an ISO-8601 timestamp and Git revision hash.

`GAME_DURATION` The duration of the game in seconds.

`GAME_ID` An integer value used as a seed for randomly-generating values used for entity features[1].

`INITIAL_INSTRUCTOR_ID` The ID of the participant which was first assigned the role of instructor in the game (in the current dataset, always *A*).

`MOVE_DELAY` This is the constant delay in milliseconds between completing a game round and the start of the next round (in addition to latency).

`ROUND_COUNT` The number of game rounds played (but not necessarily completed).

`START_TIME` The time the experiment was started, represented as an ISO-8601 timestamp

`ANNOTATOR_IDS` A comma-separated list of the unique IDs of each annotator which participated in transcribing the session data.

`EXPERIMENTER_ID` The ID of the person who conducted the experiment; A person's experimenter ID and annotator ID are equivalent.

[1] Random values are generated using a 48-bit seed which is modified using a linear congruential formula [1, pp. 9–25] from the Java class library [5]

3.2 Participant Metadata

The file `participant-metadata.tsv` contains information about each participant in a dyad, arranged in one column for each participant.

`PARTICIPANT_ID` The ID of the participant for a given column.

`INITIAL_ROLE` The given participant's initial role, either *MOVE_SUBMISSION* for instructors or *WAITING_FOR_NEXT_MOVE* for manipulators.

`SOURCE_ID` The ID of the *source* element in the XML-based utterance annotation file `utts.xml` which corresponds to the given participants' audio source: See the *Higgins Annotation Tool* XML schema [9].

`ACCENT` A description of the participant's manner of speaking using regional and language terms such as *Greek*, *Arabic*, *Scandinavian (Swedish)*, *English (General American)*, *English (Northern England)* or *South Slavic*. This information is not self-reported but rather assessed by the annotator(s).

`AGE` The age of the participant: In the dataset, this is always *Adult*, i.e. above the age of majority, which in Sweden is 18 years of age.

`DYAD_PARTNER_FAMILIARITY` A description of how familiar the participant is with their partner: *Strong* indicates close friendship or professional relationship; *Weak* indicates acquaintanceship such as being classmates or distant colleagues; *None* indicates complete strangers. This information is not self-reported but rather assessed by the experimenter/annotator(s).

`GENDER` A description of the participant's manner of speaking using gender-specific terms such as *masculine* and *feminine*. This information is not self-assessed but rather assessed by the annotator(s), being evaluated solely on the participant's speech and not on their self-identified gender.

`SAMPLING_RATE` The audio recording sampling rate in hertz.

`BIT_DEPTH` The number of bits of information in each audio recording sample.

`MICROPHONE` The microphone used to record the given participant's speech.

`AUDIO_CARD` The audio card used to record the given participant's speech.

`RECORDING_LIBRARY` The software library used for controlling recording of the given participant's speech.

`SOUND_SERVER` The sound server used managing the recording device used.

`RECORDING_SOFTWARE` The software used for writing the recorded audio to disk.

`COMPUTER` A description of the computer used by the participant during the experiment.

`DISPLAY` The visual display used by the given participant.

`GRAPHICS_CARD` The graphics card used by the given participant.

`NOTES` Additional notes about the experiment, the participant themselves, or transcription of the participant's speech.

`CONSENT_FORM` Indicates whether a signed consent form is present for the participant or not.

3.3 Event Logs

An event log of all actions in the online game is generated by the client application used by each participant: Each single event in the game is marshalled as a JSON structure using the *IrisTK* framework [10] and are written to the file `events-<PARTICIPANT_ID>.txt`, whereby participant A is the participant which first received the role of instructor, and participant B is correspondingly the participant which first received the role of manipulator. In order to facilitate processing without depending on *IrisTK*, a "canonical" event log is written as tab-separated values [3] in `events.tsv`, which is derived from the values in the event log for participant A. However, the original participant-specific event logs can still be of use, e.g. for calculating round-trip latency from the differences between times of analogous events in the participants' individual logs.

3.3.1 Records

Each row in the tabular event log file `events.tsv` represents a record that describes the state of a single entity in the game at a given time.

`EVENT` The ID of the game event the record is attributed to.

`ROUND` The ID of the game round during which the event occurred.

`NAME` A unique textual identifier of the type of event which occurred: *nextturn.request* occurs at the beginning of a new round and denotes the state of the game at the start of the new round; *selection.request* occurs when a participant selects any piece; *completedturn.request* occurs when the last-selected piece is confirmed as the correct one; *selection.rejection* occurs when the last-selected piece is confirmed as incorrect.

`SUBMITTER` The participant who initiated the event.

`ENTITY` The ID of the entity the given record describes.

`REFERENT` A Boolean value indicating if the entity the given record describes is the "target" referent, i.e. is the entity which must be correctly selected in the given round.

`SELECTED` A Boolean value indicating if the entity has been selected by the manipulator or not.

All other values represent physical features of the entity the given record describes.

3.4 Audio Recordings

Each experiment session is recorded as a WAV file using linear PCM and two channels — one channel for each dialogue participant. The two-channel files are then segmented into individual utterances and transcribed into sequences of words (tokens) using the *Higgins Annotation Tool* (HAT) [9].

When segmenting, annotators were instructed to segment audio into minimals span of uninterrupted language which denote a dialogue act in the scope of the task at hand [cf. 11]. Disfluencies and self-repair delimit segmentation boundaries only if there is a significant period of silence after the potential boundary or if the other participant takes a dialogue turn, leading the participant to respond to the other's speech act [cf. 6].

Metalanguage is a set of pre-defined labels, which are always capitalized: `ARTIFACT`, `BREATH`, `CLICK`, `COUGH`, `GASP`, `GROAN`, `GRUNT`, `LAUGHTER`, `MOAN`, `NOISE`, `PUFF`, `SIGH`, `SNIFF`, `START_SIGNAL`, `UNKNOWN`

Disfluencies are indicated with either a leading or trailing hyphen, such as *-tains* instead of *mountains* or *l-* in *big block l- top left.*

In addition being written in the original HAT XML format in `utts.xml`, the utterance data is also written as tab-separated values in `utts.tsv`:

`ROUND` The ID of the game round during which the utterance occurred.

`SPEAKER` The ID participant who made the utterance.

`DIALOGUE_ROLE` The dialogue role of the speaker in the round during which the utterance occurred, either *INSTRUCTOR* or *MANIPULATOR*.

`START_TIME` The time the utterance began in seconds from the start of the game.

`END_TIME` The time the utterance ended in seconds from the start of the game.

`TOKENS` The utterance transcription in individual tokens (words).

3.5 Miscellany

In addition to data about the state of the game and about participants' use of language during the game, there are a number of miscellaneous data sources.

Client application logs are written by the applications used by the individual participants to `system-<PARTICIPANT_ID>.log`, which may of use for debugging purposes.

Image visualization data is written to `img-info.tsv`, which describes static features used for visualizing each entity in the game for a given dyad. Position features are not included because they are dynamic and change throughout the course of the game.

Screenshots are saved at the beginning of each new round (`round-<ROUND_ID>-<TIMESTAMP>-<PARTICIPANT_ID>.png`) and also for each selection the manipulator makes (`selection-entity<ENTITY_ID>-<TIMESTAMP>-<PARTICIPANT_ID>.png`); These are stored under the directory `screenshots`.

Data structure information for each session is encoded in the Java properties file [4] `desc.properties`, which maps participant-specific information in utterance XML files to information in *IrisTK* log files. This approach allows use of these files for processing as an alternative to the tabular data files if desired.

References

[1] Donald E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. 2nd. Boston, MA, USA: Addison-Wesley, 1981.

[2] Robert M. Krauss and Sidney Weinheimer. "Changes in reference phrases as a function of frequency of usage in social interaction: a preliminary study." In: *Psychonomic Science* 1.1 (1964), pp. 113–114. DOI: 10.3758/BF03342817.

[3] Paul Lindner. *Text Media Types: text/tab-separated-values*. Internet Assigned Numbers Authority. June 1993. URL: https://www.iana.org/assignments/media-types/text/tab-separated-values (visited on 02/01/2018).

[4] Oracle Corporation. *Properties (Java Platform SE 8)*. URL: https://docs.oracle.com/javase/8/docs/api/java/util/Properties.html (visited on 02/01/2018).

[5] Oracle Corporation. *Java™ SE Development Kit 8, Update 45 (JDK 8u45)*. 2015. URL: http://www.oracle.com/technetwork/java/javase/8u45-relnotes-2494160.html (visited on 02/01/2018).

[6] Emanuel A. Schegloff. "Overlapping Talk and the Organization of Turn-Taking for Conversation." In: *Language in Society* 29.1 (2000), pp. 1–63.

[7] Michael F. Schober and Herbert H. Clark. "Understanding by Addressees and Overhearers." In: *Cognitive Psychology* 21 (1989), pp. 211–232.

[8] Todd Shore, Theofronia Androulakaki, and Gabriel Skantze. "KTH Tangrams: A Dataset for Research on Alignment and Conceptual Pacts in Task-Oriented Dialogue." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. (Miyazaki, Japan, May 7–12, 2018). Ed. by Nicoletta Calzolari et al. To appear. Paris, France: European Language Resources Association (ELRA).

[9] Gabriel Skantze. *Higgins Annotation Tool*. KTH Royal Institute of Technology. URL: http://www.speech.kth.se/hat/ (visited on 02/01/2018).

[10] Gabriel Skantze and Samer Al Moubayed. "IrisTK: A Statechart-based Toolkit for Multi-party Face-to-face Interaction." In: *ICMI '12: Proceedings of the 14th ACM International Conference on Multimodal Interaction.* (Santa Monica, California, USA, Oct. 22–26, 2012). Ed. by Louis-Philippe Morency et al. New York, NY, USA: ACM, pp. 69–76. DOI: 10.1145/2388676.2388698.

[11] Andreas Stolcke et al. "Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech." In: *Computational Linguistics* 26.3 (2000), pp. 339–373. DOI: 10.1162/089120100561737.