# NOESIS: Noetic End-to-End Response Selection Challenge

Response Space Size and Variability, Language Richness, Task Knowledge

Proposers: Dr. Lazaros Polymenakos, Dr. Chulaka Gunasekara – IBM Research

Dr. Walter Lasecki, Dr. Jonathan K. Kummerfeld – University of Michigan

## Overview

Work in response selection in previous years has focused on task-oriented dialog with the goal of collecting the necessary parameters and then issue the correct API call. The dataset was synthetically generated and geared towards evaluating the accuracy of predicting the next utterance so as to complete a dialog. Research within this setting has advanced dramatically for end-to-end systems that reach 100% accuracy, matching the performance of rule-based systems.

The challenge proposed this year aims to push the state-of-the-art of goal-oriented dialog systems in four directions deemed necessary for practical automated agents. We focus around end-to-end dialog systems that learn from chatlog data not only the parameters needed to complete the task and the correct responses, but also can deal with the following advanced conditions:

1) Natural language diversity/richness: We introduce natural language human-to-human datasets with additional human generated paraphrases.
2) Know what's right among a large number of choices: We introduce many possible candidate answers that the system has to choose from.
3) Know what's wrong when the correct answer is not included in the choices: We extend the above by introducing the possibility that the correct answer is not included in the large number of choices.
4) Knowledge grounding: We provide knowledge sources related to the goal-oriented task that can be included to improve the accuracy of the next utterance selection.

Participating systems should not be had-crafted, rule-based systems or based on hand-crafted features. Automation is the focus, so the systems have be learn directly from the provided chatlog data or leverage the additional knowledge sources provided. Participants can use the provided knowledge sources as is, or automatically transform them to appropriate representations (e.g. knowledge graphs, continuous embeddings, etc.) that can be integrated with end-to-end dialog systems so as to increase response accuracy. For training and evaluation we introduce two new datasets and we center the subtasks in a progression of capabilities/conditions the systems will evaluated on, so that useful comparisons and baselines can be drawn.

## Task Description and goals

The challenge focuses on goal-oriented dialog. Two datasets are provided:

1. Flex Data: Student – Advisor dialogues for the purpose of guiding the student to pick courses that fit not only their curriculum, but also personal preferences about time, difficulty, career path, etc. Additional knowledge base about courses and possible (but

not all) personal preferences will be provided. The data also includes paraphrases of the sentences and of the target responses.
These are play-acted data following a set of possible selections for courses and for a progression of advisor dialog acts.

2. Ubuntu Dialog Corpus: A new version of disentangled Ubuntu IRC dialog will be provided. The purpose is to solve an Ubuntu user's posted problem – two-party dialogues are provided. Additional knowledge will be provided in the form of manual pages.

There are 5 subtasks described below. A participant may participate in one, several or all the subtasks:

| **Subtask** | **Evaluated on** | |
|---|---|---|
| | **Ubuntu dataset** | **Flex dataset** |
| 1. Baseline – Select the next utterance from given candidate set (candidate pool < 100) | ✓<br>The set will contain between 1 option that is correct and 99 options that are incorrect (for a total of 100). | ✓<br>The set will contain between 1 option that is correct and 99 options that are incorrect (for a total of 100). |
| 2. Select the next utterance from a large global pool of candidates (candidate pool > 10000) | ✓<br>A large pool of candidates (over 10000) will be provided to pick the next utterance from.<br>The increased number of candidates will challenge the *logical capability* of dialog models. | |
| 3. Select the next utterance with the set of paraphrases. | | ✓<br>The set will contain between 1 and 5 options that are correct, and 95 - 99 options that are incorrect (for a total of 100). We provide multiple correct options by using paraphrases (note: the correct options in a set may include the original utterance |

| | | |
|---|---|---|
| | | we collected or may be only paraphrases). |
| 4. Select the next utterance with a candidate pool which might not include the correct next utterance for some instances (candidate pool <100). Only one answer is correct, no paraphrases will be provided. | ✓ | ✓ |
| 5. Select the next utterance with a model which incorporate external knowledge (candidate pool < 100). The external knowledge base will be provided. | ✓ (Ubuntu manual pages) | ✓ (Curriculum Related Database) |

**Ubuntu related subtasks:** The training data will include over 100000 complete conversations, and the test data will contain 1000 partial conversations. Each dialog will have a minimum of 3 turns.

**Flex related subtasks:** The training data will be based on 500 conversations. We will provide the training data in two forms. First, the 500 conversations with a list of paraphrases for each utterance. Participants are welcome to use this data in any way and are encouraged to explore training methods. Second, we will provide 100,000 partial conversations that are of the same format as the test set; a partial conversation, and a set of 100 options for the next sentence. The test data will consist of 500 instances, where each instance is a partial conversation, and a set of options for the next utterance, including between 1 and 5 that are paraphrases of the true next utterance. We will construct these 500 instances by taking 100 dialogues and cutting them off at five different points. To make the five instances from a dialogue different, we will use paraphrases. The sets of next utterance options will also be distinct from the partial conversations we provide. Each set will contain 100 options, of which between 1 and 5 are correct (the number will be chosen randomly). The incorrect options will be chosen by

randomly sampling other turns in the data and then randomly choosing how many paraphrases to include (between 1 and 5 for each).

## Evaluation

For each test instance, participants will return a set of 10 choices from the set of possible follow-up sentences and a probability distribution over those 10 choices. For the competition metric we will consider the choices that cover 90% of the distribution, and compute an F-score as the harmonic mean of precision and recall:

Precision = (number of correct sentences selected) / (total number of sentences selected)

Recall = (number of correct sentences selected) / (total number of correct sentences in all sets)

We will also do analysis of performance with other metrics, such as:

- F-score on the top N choices, where N is the true number of correct options in the response set (R-precision).
- Evaluation of systems for their ability to provide just one correct paraphrase for the next adviser utterance by considering the rank of the first correct paraphrase returned by the system (and ignoring all the other paraphrases).

## Baselines

We will implement and evaluate several simple and Neural network baselines that rank the candidate utterances.

## Timeline

| | |
|---|---|
| Mar – May 2018: | Track preparation |
| Jun 1 – Sep 9, 2018: | Development phase (14 weeks) |
| Sep 10 – Sep 24, 2018: | Evaluation phase (2 weeks) |
| 1 Oct 2018: | Objective evaluation results are released |
| 8 Oct 2018: | Human evaluation results are released |
| Oct or Nov 2018: | Paper submission deadline |
| Spring 2019: | DSTC7 special session or workshop |

# Appendix 1: Flex Data

## Example partial dialogue:

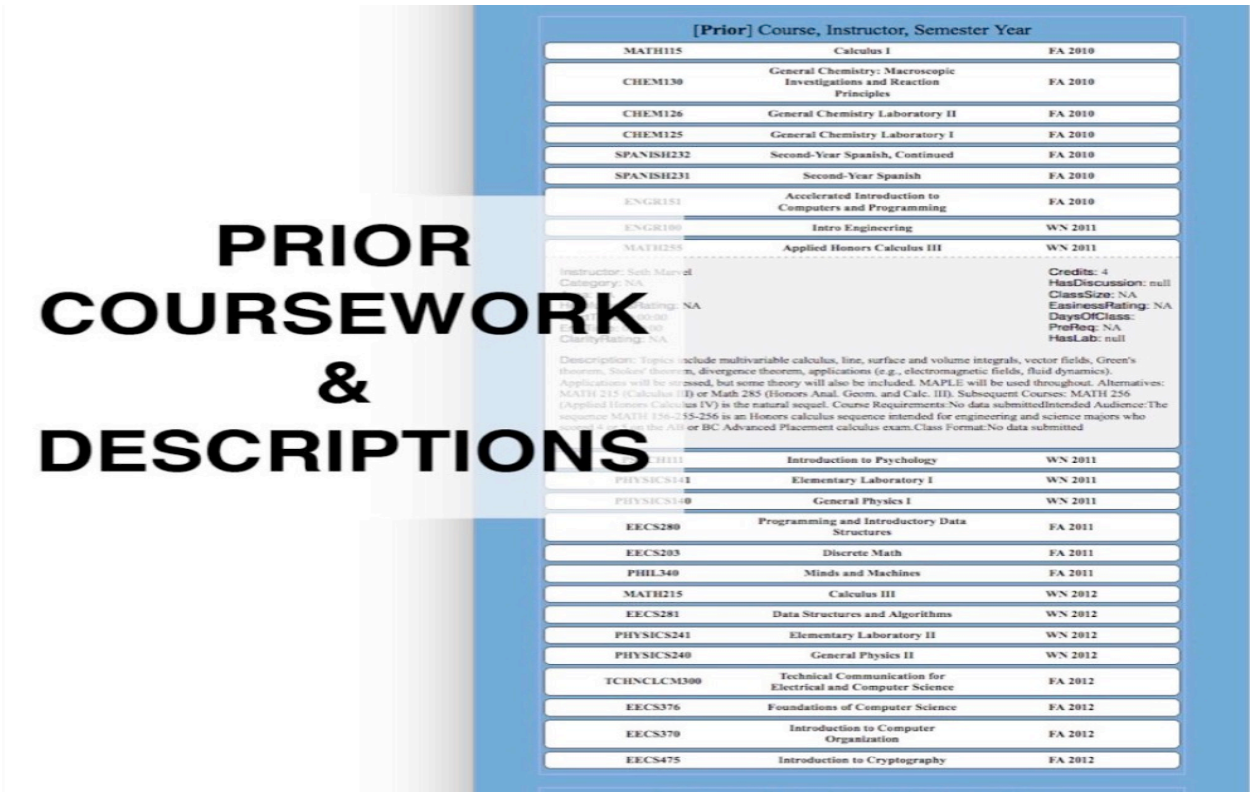
*ADVISOR | Hi! What can I help you with?*
*STUDENT | Hello! I'm trying to schedule classes for next semester. Can you help me?*
*STUDENT | Hardware has been an interest of mine.*
*STUDENT | But I don't want too hard of classes*
*ADVISOR | So are you interested in pursuing Electrical or Computer Engineering?*
*STUDENT | I'm undecided*
*STUDENT | I enjoy programming but enjoy hardware a little more.*
*ADVISOR | Computer Engineering consists of both programming and hardware.*
*ADVISOR | I think it will be a great fit for you.*
*STUDENT | Awesome, I think that's some good advice.*
*STUDENT | What classes should I take to become a Computer Engineer?*
*ADVISOR | You haven't taken EECS 203, 280, and 270, so it may be in your best interest to take one or two of those classes next semester*
*STUDENT | Ok. Which of those is in the morning. I like morning classes*

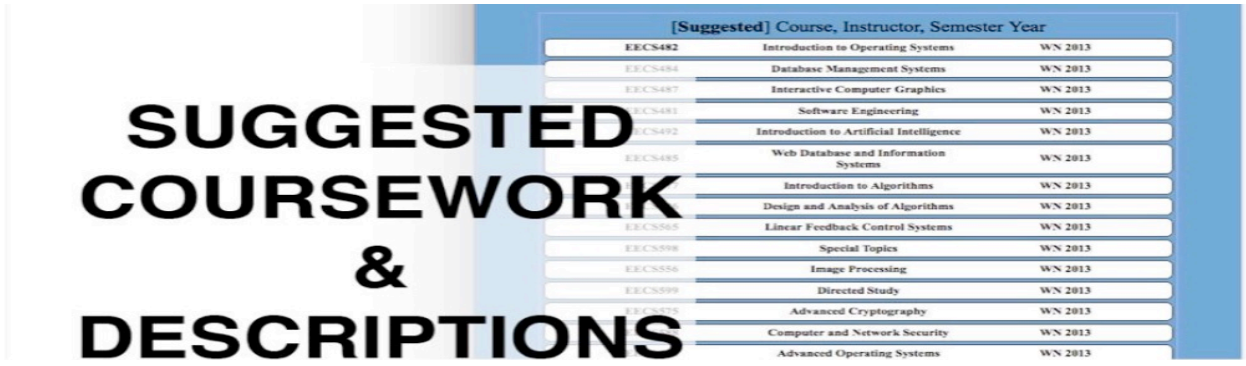
## Example Candidate set:

Twenty next utterance options, correct ones shown in bold:

- *Is there anything else I can help answer?*
- ***They have not released the plans for next semester yet.***
- *Do you have an interest in this class?*
- *Do you find this class interesting?*
- *Does this course interest you?*
- *It wouldn't be smart to combine 381 with another EECS course, unless you like to stay up late*
- *It wouldn't be in your best interest to choose combining 381 with another EECS course, unless*
- *you do well staying up real late.*
- *it would not be a wise choice to combine 381 with another EECS course, unless you like to burn that midnight oil*
- *Its not wise to combine a EECS course with 381, unless you want to stay awake all night.*
- *You might over-extend yourself by taking another EECS course combined with 381.*
- ***They have not released the schedule for next semester yet.***
- *Do you have any interest for this course?*
- *Are you interested in this course?*
- *Taking both 381 and another EECS would not be a wise choice.*
- *Registering for EECS370 with EECS281 is a good choice.*
- *Do you have any other questions for me?*
- *Does this class interest you?*
- ***They have not released the schedule for next term yet.***
- *I wouldn't recommend taking 381 at the same time as any other EECS course - you'll be up all night working.*
- *questions you have?*

## Knowledge sources




# Appendix 2: Ubuntu data version 3

## Example partial dialogue

*[13:11] <user_1> anyone here know memcached?*
*[13:12] <user_1> trying to change the port it runs on*
*[13:12] <user_2> user_1: and ?*
*[13:13] <user_1> user_2: I'm not sure where to look*
*[13:13] <user_1> !*
*[13:13] <user_2> user_1: /etc/memcached.conf ?*
*[13:13] <user_1> haha*
*[13:13] <user_1> user_2: oh yes, it's much simpler than I thought*
*[13:13] <user_1> not sure why, I was trying to work through the init.d stuff*

## Example Candidate set:

Ten next utterance options, correct ones shown in bold:

*<user_2> user_1: but yea the processor gets low*
*<user_2> user_1: I dunno.. I just want to send an email to say foo@limcore.com and I don't care to read any reply*
***<user_2> user_1: that would be the second place to look***
*<user_2> user_1: i mean the number of updates?*
*<user_2> user_1: cause gnome is more than tolerable in slack, but it's friggin' blazing in Ubuntu*
*<user_2> user_1: how about properties?*
*<user_2> user_1: its not there*
*<user_2> user_1: is your adapter working properly?*
*<user_2> user_1: search for it in synaptic*
*<user_2> user_1: oops wrong channel*

## External knowledge (An example from Linux manual pages):

### Name

dos2unix - DOS/MAC to UNIX text file format converter

### Synopsys

dos2unix [options] [-c convmode] [-o file ...] [-n infile outfile ...]

Options:

[-hkqV] [--help] [--keepdate] [--quiet] [--version]

### Description

This manual page documents dos2unix, the program that converts plain text files in DOS/MAC format to UNIX format.

### Options

The following options are available:

**-h --help**
Print online help.
**-k --keepdate**
Keep the date stamp of output file same as input file.
**-q --quiet**
Quiet mode. Suppress all warning and messages.
**-V --version**
Prints version information.
**-c --convmode convmode**
Sets conversion mode. Where convmode is one of: **ASCII, 7bit, ISO, Mac** with ASCII being the default. Simulates dos2unix under SunOS.
**-o --oldfile file ...**
Old file mode. Convert the file and write output to it. The program default to run in this mode. Wildcard names may be used.
**-n --newfile infile outfile ...**
New file mode. Convert the infile and write output to outfile. File names must be given in pairs and wildcard names should NOT be used or you WILL lose your files.