

---

# Alana v2: Entertaining and Informative Open-domain Social Dialogue using Ontologies and Entity Linking

---

Amanda Cercas Curry, Ioannis Papaioannou,  
Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu  
Ondřej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser and Oliver Lemon  
The Interaction Lab, Department of Computer Science  
Heriot-Watt University, Edinburgh, EH14 4AS, UK  
{ac293, i.papaioannou, as247, sa201, is33, xx6}@hw.ac.uk  
{o.dusek, a.eshghi, i.konstas, v.t.rieser, o.lemon}@hw.ac.uk

## Abstract

We describe our 2018 Alexa prize system (called ‘Alana’) which consists of an ensemble of bots, combining rule-based and machine learning systems. This paper reports on the version of the system developed and evaluated in the semi-finals of the 2018 competition (i.e. up to 15 August 2018), but not on subsequent enhancements. The main advances over our 2017 Alana system are: (1) a deeper Natural Language Understanding (NLU) pipeline; (2) the use of topic ontologies and Named Entity Linking to enable the user to navigate and search through a web of related information; rendering Alana in part an interactive NL interface to linked information on the web; (3) system generated *clarification questions* to interactively disambiguate between Named Entities as part of NLU; (4) a new profanity & abuse detection model with rule-based mitigation strategies; and (5) response retrieval from Reddit. We also present several ablation studies that measure the performance contributions of specific features (e.g. use of Ontology-bot, Reddit-bot, rule-based systems, etc). We find that these features increase overall system performance. Our final score, namely averaged user ratings over the whole semi-finals period, was 3.4. We were also able to achieve long dialogues (average around 11 turns and 2.20 minutes) during the semi-finals period.

## 1 Introduction

In this paper, we describe our entry to the 2018 Alexa Prize challenge semi-finals, a socialbot called Alana. Our system is based on an ensemble of different task/topic-specific bots, combining rule-based and machine learning systems. We focus mostly on the improvements we made over our last year’s entry (Papaioannou et al., 2017a,b).

Our overall vision was to create an informative and engaging social chatbot that aims to keep users interested and enjoying a spoken interaction on topics of their choice for as long as possible. Our overarching inspiration for this vision is a mixture of topic-related chat, finding out about the user, and sharing amusing facts, jokes, stories, and items of news. Given that the system has better than human access to information on the web such as Wikipedia and news articles, our system also provides an engaging and interactive way to explore the web/news according to user preferences. As such, our social chatbot was designed to have the following behaviour:

1. It should be able to engage in **open-domain topic-based** conversations, to minimise responses such as “I don’t know what you mean” or “I can’t answer that”.

2. The replies should sound **natural** and **non-repetitive**.
3. The replies should be **engaging** and **informative**, i.e. stimulate further conversation.

In order to handle the specific topic-related aspects of the Alexa challenge (e.g chat about movies, music, news, baseball, celebrities, etc.), we extend our previous system with ontologies and with topic-tagged data, such as sub-Reddits.

## 2 System Design and Architecture

### 2.1 Overall System Structure

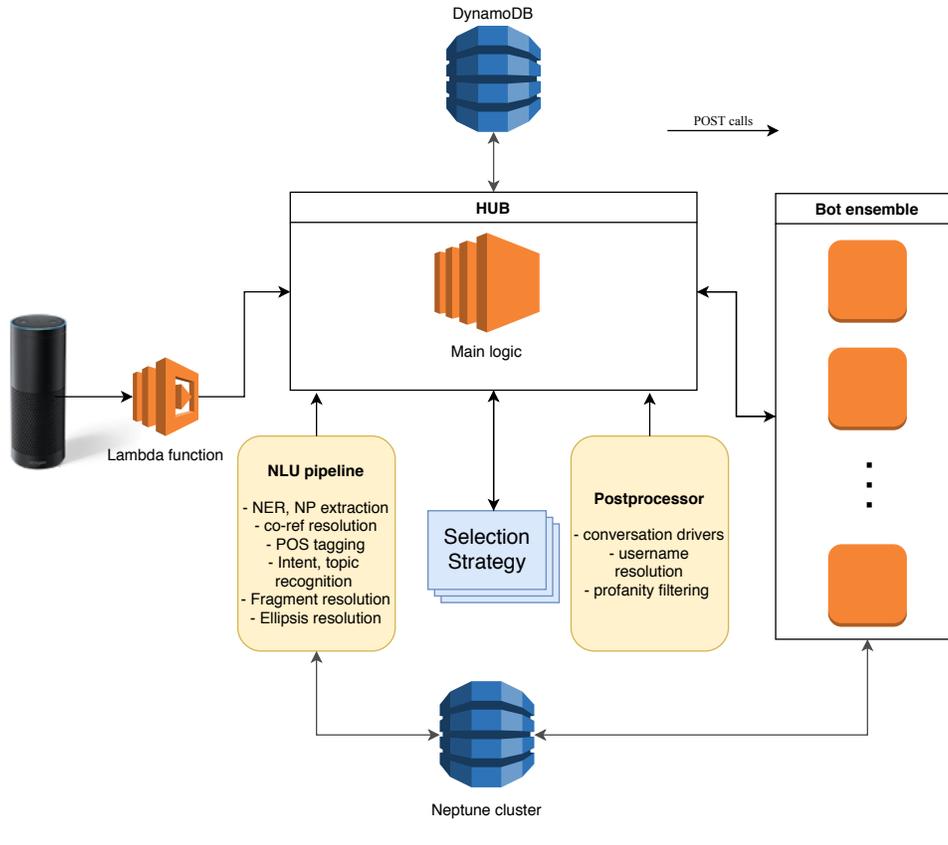


Figure 1: Alana architecture

*Alana* uses a modular design as shown in Fig 1. When the user talks to their *Amazon Echo* device, a new session begins and an event object reaches the *lambda function*, which contains the text representation of the user’s utterance, along with several other high-level metadata (such as tokenised confidence scores). The *lambda function* then forwards this information to an EC2 instance (henceforward called the *Hub*), which provides the main logic of the system. The *Hub* is the overarching component of our system, which ties together all the other modules, and dictates the flow of information between them. The *Hub* is also the only module in the architecture that has access to the DynamoDB database.

This initial information updates the initial *state object*, which stores information about the current context (see section 2.2). At this stage a list of all (or n-last) previous states is pulled from the database which along with the current state object are then forwarded to the NLU module (described in section 3.1.1), in order to be enriched with annotations for that specific user utterance.

After the NLU pipeline step has been completed, and the state object is updated with the annotations, the *dialogue history* (*list of states of the conversation that the system has encountered so far for*

the current session), current state and user attributes (see section 3.3) are being forwarded to the ensemble of bots, as described in sections 3.2 and 3.3. Each bot then generates one or multiple candidate responses, which are then collected back by the Hub. Since the Hub is the only component having access to the database, whatever information each bot needs to be retained across turns, it is also returned to the hub as part of its response, which is then incorporated in the *object state* inside the *bot\_states* attribute.

Once all candidate responses are collected, a *selection strategy* is being applied in order for the contextually best response to be selected. Currently the selection strategy is defined by a Priority Bot List, which states which bots should handle the current turn, relying on probabilistic decisions with hand-crafted weights. We plan to augment this with a trained Ranking function (Papaioannou et al., 2017a) in future work. The Priority List is further discussed in section 3.1.5.

On certain occasions, a bot (or multiple bots) might be required to handle the current turn’s response (e.g. during a multiturn sub-dialogue), in which case it sends to the *Hub* a “lock” request, which overrides the bot priority list.

Once a response is selected, it is then post-processed, by occasionally injecting the user’s name (if known and appropriate) in grammatically appropriate places in the response, as well as adding conversational drivers, in order to keep the flow of the dialogue (see *Coherence bot* in section 3.3). Finally, the fully updated *state object* is then appended to the database under the same session. In order to conform to the time limits set out in the competition rules, each component includes a time-out limit of 5 seconds.

## 2.2 Dialogue State

Supporting an interesting and engaging conversation requires a detailed representation of the current state of the dialogue. For this reason, we designed a specific *context representation* for the Alana system aimed at integrating and retaining fundamental information related to the state of the conversation. The structure of the state in each turn is shown in Figures 2.

```
{
  'session_id': ...,
  'timestamp': ...,
  'state': {
    'turn_no': ..., # turn number
    'last_bot': ..., # bot that produced the previous response
    'input': { # ASR input
      'text': ..., # 1-best plain text
      'avg_conf_score': ..., # average confidence score
      'raw': [...], # raw ASR input (words with scores)
    },
    'nlu': { # NLU annotation
      'annotations': {
        'intents': {...},
        'ner': {...},
        'processed_text': {...},
        'profanity': {...},
        'postag': {...},
        'sentiment': {...},
        'topics': {...},
        ...},
      'modules': [...], # list of modules applied
      'processed_text': {...} # input after normalization & ellipsis resolution
    },
    'response': {<bot_name>: <response>}, # the selected bot's response
  },
}
```

Figure 2: The structure of Alana’s dialogue state (green box in Figure 3).

The *state object* encapsulates all the information needed by the system to produce an appropriate response for the user in each turn. In the following, we briefly describe some of its main components:

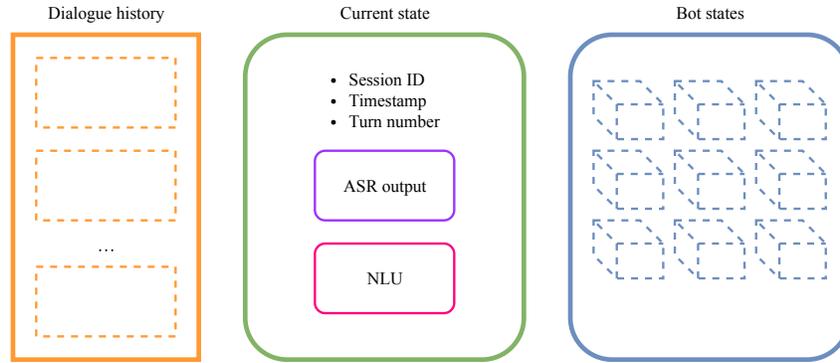


Figure 3: A schematic of the context representation used in Alana (green box corresponds to Figure 2) The dotted boxes under Dialogue History represent all the states of the conversation that the bot has encountered so far for a specific session.

- *Session ID*: identifier of the conversation;
- *Timestamp*: moment in time in which this turn started;
- *Input*: structured representation of the ASR output provided by the Amazon’s ASR model. It is composed of the following elements:
  - *User utterance (text)*: raw text coming from the Amazon ASR;
  - *Average confidence score*: average confidence score for the top ASR hypothesis;
  - *Tokenised confidence scores (raw)*: confidence score associated with each token in the top ASR hypothesis.
- *NLU*: annotations generated by the NLU pipeline (see Section 3.1.1) for the current user utterance.
- *Response*: The system’s response utterance

Two other objects accompany the current state in the Alana context representation (see Figure 3):

- *Dialogue history*: This is a simple list of state objects for all past turns in the conversation.
- *Bot states*: Each bot’s internal state, used to keep track of dialogue progress and previously completed actions within the particular bot, such as multi-turn News bot features or list of topics previously discussed for the Coherence bot.

This context is compiled and updated as the system moves through each dialogue turn (see section 2.1) and provides a common knowledge base for every module of the system.

### 3 Alana Services and Updated System Components

In this section, we focus on the new system components developed for the 2018 Alana version 2:

- New services used by the individual bots in the ensemble or the main system hub, as well as additional monitoring tools (Section 3.1),
- Newly introduced bots in the ensemble (Section 3.2).

We also include a brief summary of bots that were already part of our 2017 Alexa Prize finalist system Alana version 1 (Papaioannou et al., 2017a) in Section 3.3, noting any updates since the previous version.

#### 3.1 Bot services

Here we describe new services available to all bots in the ensemble: NLU pipeline, entity linking, clarification, and the priority list.

### 3.1.1 New NLU pipeline

A fundamental component of our updated architecture for 2018 is represented by a Natural Language Understanding (NLU) pipeline that enhances the current user utterance with specific *annotations*. The NLU service is responsible for annotating a given text by applying a pipeline  $P$  conditioned on the conversation state  $S$  previously defined in Section 2. The NLU service is equipped with a set of *modules*  $\mathcal{M} = \{M_1, M_2, \dots, M_n\}$ . From  $\mathcal{M}$  a pipeline  $P$  can be created. The pipeline is outlined as a list of *module units*. A module unit is defined as a list of modules that can be executed concurrently. Each module unit is applied sequentially and receives the state object generated in the previous step of the pipeline. In this way, the behaviour of each NLP module is conditioned on the state of the conversation. This is a fundamental capability for a conversational system that needs to take into account different information in a conversation such as topics, mentioned entities, and intents associated with the current utterance.

The designed NLU service is composed of several state-of-the-art components such as:

1. *Truercaser*: we adopt a trained language model<sup>1</sup> to determine the correct capitalisation for each word in the user utterance. This is fundamental in order to boost the performance of the Named-Entity Recognition (NER) model.
2. *Contextual preprocessor*: this is a custom module able to transform the user utterance by using contextual information. We resolve elliptical yes/no user replies (e.g. *Yes, No, I don't* etc.) by transforming their utterance into a full sentence in context (e.g. if in the last turn the system asked “*Do you like tea*”, and the user responds with a “*Yes*”, the response will be transformed into “*Yes I do like tea*”). Indirect user questions such as “*I don't know who X is*” are transformed into direct questions such as “*Who is X*”.
3. *POS tagger*: we rely on the *MorphoDiTa* Part-of-Speech tagger (Straková et al., 2014) to extract POS tags for the user utterance.
4. Regex-based intent recogniser: multiple patterns associated with each intent are represented by means of regular expressions and matched against the current user utterance.
5. Named Entity Recogniser Ensemble: we designed an ensemble of NER models composed of *SPaCy* NER<sup>2</sup> and *Stanford NER*.
6. Sentiment Analyser: we used the Sentiment Analyser provided by the NLTK library<sup>3</sup>
7. Entity Linker: we exploit a custom version of the Fast Entity Linking service (see Section 3.1.2 for details) to find entity mentions in the the user utterance and link them to entities in the Wikidata<sup>4</sup> knowledge base.
8. Entity Topic Classifier: we exploit the recognised entities in the knowledge base to determine the topic of the conversation. Specifically, each entity has an associated set of properties that can be taken into account to extract a set of candidate topics for the current turn. In particular, we stick to the previous topic if it is among the candidate topics for the current turn otherwise we randomly select one if we do not have a topic set in the previous turn.
9. Anaphora Resolution: we exploit both the annotations generated by the NER ensemble and the annotations generated by the entity linker to resolve coreference - see Fig. 4 for an example. When a pronoun is detected in the user utterance, we try to associate to it the last mentioned entity in either the user utterance or the bot utterance. We exploit the information coming from the entity linker in order to provide a better resolution strategy that takes into account the gender (if available) or the type of the entity to match it to the most appropriate pronoun.

### 3.1.2 Entity Linking

Once we have a knowledge base available, in order to retrieve additional information associated with the entities that can be found in it, the system should be able to map a given *surface form* in the user

<sup>1</sup><https://github.com/nreimers/truercaser>

<sup>2</sup>[https://spacy.io/models/en#en\\_core\\_web\\_lg](https://spacy.io/models/en#en_core_web_lg)

<sup>3</sup>[https://www.nltk.org/\\_modules/nltk/sentiment/vader.html](https://www.nltk.org/_modules/nltk/sentiment/vader.html)

<sup>4</sup>[https://www.wikidata.org/wiki/Wikidata:Main\\_Page](https://www.wikidata.org/wiki/Wikidata:Main_Page)

utterance to an entity in the related knowledge base. This task is known as *Entity Linking* and it is defined by Shen et al. (2015) as follows:

“Given a knowledge base containing a set of entities  $E$  and a text collection in which a set of named entity mentions  $M$  are identified in advance, the goal of entity linking is to map each textual entity mention  $m \in \mathcal{M}$  to its corresponding entity  $e \in \mathcal{E}$  in the knowledge base.”

This is an important task in dialogue systems that needs to understand mentions to specific entities in the user utterance: it is fundamental if we want the system to be able to support a coherent conversation with the user about a specific topic. For instance, we want the system to be able to detect when the user is talking about a movie in order to retrieve additional information that can be interesting for the user (or to talk about similar movies).

In order to tackle this task we use the *Fast Entity Linking (FEL)* system (Blanco et al., 2015; Pappu et al., 2017). This tool can associate to a given surface form  $s$  in the user utterance a set of candidate entities  $\mathcal{S}_s$ . The linker generates for each candidate entity  $e \in \mathcal{S}_s$  a confidence value  $\sigma(e)$  that represents the probability that the candidate entity  $e$  is an appropriate candidate for the surface form  $s$ . We decided to extend the original entity linker in 3 different ways:

1. *contextual filter*: we filter out irrelevant candidate mentions in a specific turn by exploiting Wikidata properties associated to the entities. We retain only the candidate entities for a given surface form that satisfy specific entity types and property values. Specifically, according to the current topic of the conversation, the entity linker filters out entities that are not considered *coherent* with it. Currently, the entity linker supports the following topics: movies, books, music, video games, and sports.
2. *dependency parser filter*: we rely on syntactic information extracted using the *Neural Stanford Dependency Parser* (Chen & Manning, 2014). In particular, we keep spans of text that contain possible entities, based on the dependency label of the head. The rationale is to focus on spans that are either the object, the noun subject of the main verb of the user utterance, or a single noun phrase (elliptic sentence). We implement that by ignoring all the candidate spans extracted by the entity linker that *do not* satisfy the following properties:
  - **(in)-direct object**: the dependency label of the head of the span of text is equal to one of  $\{obj, xcmp, ccomp, nmod\}$ ;
  - **noun subject**: the dependency label of the head of the span is equal to  $nsubj$  and its POS tag is *NOUN*;
  - **single NP**: the dependency label of the head of the span is equal to  $root$  and its POS tag is *NOUN*.
3. *candidate threshold filter*: we apply a thresholding procedure on the candidate entities for each surface form. In particular, in order to filter out entities that are not relevant we use two different threshold techniques in sequence:
  - (a) *maximum annotation threshold*: we indicate with  $\mu_s$  the candidate entity associated to  $s$  with the maximum score  $\sigma(e)$ . We ignore a surface form  $s$  in the user utterance if  $\sigma(\mu_s) \leq \gamma_a$  the threshold value.
  - (b) *maximum candidate threshold*: we ignore a candidate annotation  $e \in \mathcal{S}_s$  if  $\sigma(e) \leq \gamma_c$ .

It is worth noting that by default FEL maps each entity to the corresponding Wikipedia page. So, we query the Wikidata endpoint in order to retrieve the equivalent Wikidata entity. There are two edge cases to this procedure: 1) the Wikidata page is not available because the Wikipedia page redirects to a disambiguation page; 2) the Wikidata page is not available through the Wikidata endpoint. In the first case, we ignore the mention. In the second case, we rely on the *DBpedia* endpoint<sup>5</sup> to retrieve the equivalent entity in the DBpedia knowledge base and, through the property *owl:sameAs*, we search for the related entity in the Wikidata knowledge base.

### 3.1.3 Clarification Service

Being able to retrieve specific entities in the user utterance is not always enough to achieve a coherent and engaging conversation. Sometimes for a given surface form there are multiple candidate entities

<sup>5</sup><http://dbpedia.org/sparql>

in the knowledge base. This preliminary step in the Entity Linking literature is called *Entity Spotting*. Instead of relying on a disambiguation procedure that completely ignores the user, we decided to implement an *interactive clarification module*. Its purpose is to resolve the ambiguity of an entity interactively by posing a *clarification question*. This module is composed of several steps that are reported as follows:

1. Identification of ambiguous candidates;
2. Generation of a clarification question;
3. Analysing the user answer;
4. Clarifying the entity and continuing the dialogue.

If we still have multiple candidates after applying the above-mentioned filters, we generate a *clarification question* that will ask the user to choose between two possible candidates (e.g. ‘Do you mean Blade Runner the sci-fi movie or the South African Paralympic athlete?’; see also Fig. 4). Specifically, we first order the candidates according to the confidence score generated by FEL and then we select the first two entities with the highest score. To generate the question for the user, we retrieve from the Wikidata endpoint useful information for them and we use them to fill specific question templates. In the last step we analyse the user response to the clarification question. In order to do that, we use an  $n$ -gram pattern matching, that will compute all the possible ways to identify an entity, based on how it is described in the clarification question (for example ‘Do you mean *The Expanse the American science fiction television series, 2015*, based on novels by James S. A. Corey or *The Expanse the series of space opera novels by James S. A. Corey?*’). Once we clarify the entity, we refine the entity linker annotations in the NLU and we restore the dialogue context to the turn preceding the clarification.

### 3.1.4 Individual Bot Classifiers

Sometimes individual bots might generate responses that are less suitable than others. For example, they provide a very generic response, the entity disambiguation is erratic, the response is about the wrong entity mention in the user utterance, and so on. In an effort to avoid outputting such responses, we built bot-specific classifiers given the context of the dialogue.

In particular, we use a subset of dialogues collected during the 2017 competition, and extract the system response generated by a single bot, the previous user utterance, the last sentence of the previous system response and the name of the previous bot. Then we manually annotate each response as either *appropriate*, *inappropriate* or *potentially appropriate*. We then train a simple neural classifier that comprises BiLSTM encoders for each of the extracted sequences, the concatenated output of which is then fed to a multi-layer feed-forward network (FFN) with ReLU activation functions.

So far, we have deployed a classifier for the Wiki-bot (see Section 3.3). We manually annotated 880 bot responses from 2k dialogues, making sure they originate from dialogues with a balanced set of ratings. We have also automatically annotated another 1k bot responses (from a different set of 2k dialogues) using the following heuristics: we label as *appropriate* if the next user utterance is affirmative to a bot response that asks a user if they want to hear more from Wikipedia about the current entity mention. We label as *inappropriate* if the user ends the conversation immediately after the bot response and gives a low rating ( $\leq 2$ ). We partition the dataset to 80-10-10 train-dev-test splits and train using ADAM (Kingma & Ba, 2014). All BiLSTMs have 1 layer, and the final FFN 5 layers. We use pretrained GloVe embeddings (Pennington et al., 2014) for all encoders. The accuracy of the classifier is 88.18% on the dev set.

### 3.1.5 Response Selection

Our current selection strategy is defined by a bot Priority List, which dictates which bot should handle the current turn. A bot will only produce a response given a particular user intent, so the turn will be handled by the bot with the highest priority that actually produces a response. The current priority list is as follows, in order:

- Profanity bot
- Fact+Joke bot

- Weather bot
- Persona
- Ontology bot
- Reddit bot
- News bot
- Wiki bot
- Evi

Eliza and Coherence always have the lowest priority, as it is used as a “safety net” to catch utterances for which no other bot has a candidate response, hence they are not listed in the priority list. In addition, bots are able to “lock” onto a turn, for example News bot will lock if the user would like to hear the rest of a particular piece of news.

We plan to augment this selection strategy with a trained ranking function Papaioannou et al. (2017a) in future work, as described in section 5.3.

### 3.1.6 Fail-safe Mechanisms and Monitoring

To maximize availability, the system is designed to be very robust and to provide a reply to the user even under the following adverse circumstances:

- *Overload/failure of specific bots in the ensemble:* Since all bots are queried for each user input (see Section 2), in many cases multiple bots are able to respond. Therefore, if one or more bots are overloaded or faulty, other bots can handle many queries that would normally be handled by the incapacitated bot(s), albeit providing a lower-quality response. The coherence bot (see Section 3.3) is designed to always provide a reply and Eliza (see Section 3.3) will generate generic replies to many common queries, which means that the system as a whole is able to provide an output from one of the bots in most cases.
- *Entity linker failure:* The entity linker is a separate module used by the NLU as a service (cf. Sections 3.1.1 and 3.1.2). Therefore, in case it fails, the NLU is still able to provide the remaining annotations, which can still be used to provide a reply to the user. The ontology bot relies directly on linked entities, but other bots are in general able to produce responses even if entity linking is not present.
- *NLU overload/failure:* In case the whole NLU is overloaded or fails, the system employs a stalling tactic, responding that it did not hear the user correctly and asking them to try again. If this happens only once or twice throughout the dialogue, it can sustain the conversation while the system is overloaded, at the cost of a slightly worsened user experience.

The combination of these fail-safe mechanisms allowed us to sustain a 99% system uptime over the course of the semifinal period.

In addition to the fail-safe mechanisms, we implemented the following two monitoring and error reporting tools:

- *Continuous test requests to system components:* Every 10 minutes, all components of the live system (the hub, all bots, NLU, entity linker) are sent a test query that checks the functionality of the particular component. In case of a timeout or failure, the monitoring system sends an alert to the developers by email.  
Due to ongoing problems with the Neptune cluster used for entity linking and linking concept generation (see Section 3.1.2 and Section 3.2.1) where the cluster suddenly became unresponsive and had to be restarted, we also implemented an auto-restart feature using the Amazon Neptune API <sup>6</sup>: certain parts of the bot infrastructure can be automatically restarted if a timeout is encountered. This is currently in use for the Neptune cluster only.
- *Daily error reporting:* We implemented a script that generates a summary of all errors in the logs for the main hub and the NLU component, ordered by number of occurrences. These lists are emailed daily to system developers, thus providing an overview of problems in the system.

---

<sup>6</sup><https://boto3.readthedocs.io/en/latest/reference/services/neptune.html>

Both of these components allow us to maintain a reasonable system performance over time.

## 3.2 New bots in Alana v2

Here we describe the new bots developed for Alana v2: the Ontology, Abuse mitigation, and Reddit bots.

### 3.2.1 Ontology bot

**Contextualised Linked Concept Generator** An engaging conversation with a user can be obtained if the dialogue system is able to stay on topic and propose novel and interesting topics of discussion in order to prevent the conversation from getting stuck. The Wikidata knowledge base provides us with a large amount of information that, if exploited in a proper way, can allow the system to derive interesting facts that can drive the conversation forward. We exploit the *Amazon Neptune* service in order to store the entire Wikidata knowledge base and some fragments of the *DBpedia*<sup>7</sup> knowledge base.

We rely on this intuition in order to develop a core module of our dialogue system called the *Contextualised Linked Concept Generator*. In particular, every time the user mentions an entity, we try to discover interesting connections in the Wikidata graph that relate the mentioned entity with another one in the knowledge base which belong to the same domain. By leveraging SPARQL queries, we design different strategies to generate linked concepts for a given entity. Specifically, we define as *source entity*  $e_s$ , the entity mentioned in the text that has the highest score generated by the entity linking system. According to the type of the source entity, we activate different SPARQL queries that allow us to connect the source entity to another entity through a multi-hop procedure that involves domain-specific properties. In particular, for each topic, we rely on specific properties as a *bridge* that we can use to connect the source entity with plausible *linked entities*. From a set of possible linked entities  $\mathcal{L}_{e_s}$ , we select an entity  $e_l$  at random and we leverage the link between  $e_l$  and  $e_s$  to generate a response for the user by means of a template-based response generator.

Finally, the response for the user is generated using templates that involve all the hops connecting  $e_s$  to  $e_l$ . For instance, in the movie domain, if the source entity is "Pulp Fiction", the system will find "The Unborn" as a linked entity which is connected to the source entity through "Kathy Griffin". For example, here is an output template for the movie domain, using linked entities:

"Yes. How great is Pulp Fiction? I guess you know that one of the lead roles in Pulp Fiction was played by Kathy Griffin. The old movie The Unborn also starred Kathy Griffin. So, what's another movie like Pulp Fiction that you enjoyed?"

**Entity Explanations** The linked concept generation procedure is able to find links between the entities mentioned during the conversation allowing the user to learn novel relationships between them. Due to the randomness involved in the process, the linked concepts generated can be completely unknown to the user. This can be a problem because it may stall the conversation if the user says that they do not know the linked entity.

In order to cope with this problem we define an explanation mechanism that exploits the Linked Open Data information associated to each entity in order to provide a sound definition for them. In particular, according to the entity type (associated with the property *instance of*) or to some properties (e.g. *occupation*), we retrieve specific data that are used later on to fill type-specific templates used to generate the explanation. This feature allows us to provide domain-specific information associated to a given entity that can be extracted from the Wikidata knowledge base.

See Fig. 4 for an example of Entity Explanation.

**Ontology Bot architecture** The Ontology bot is composed by different topic-specific modules. Each of them can be divided in 3 main components:

- Response generator: it is the topic-specific component responsible for handling specific intents and calling internal components of the bot;

---

<sup>7</sup><https://wiki.dbpedia.org/>

- **Ontology manager:** it is responsible for the interaction with the Neptune cluster. It has domain-specific SPARQL queries that allow to provide information related to the entities as well as the linking concepts described above;
- **Driver generator:** it represents a domain-specific component able to exploit the semantic annotations retrieved from the knowledge base to generate a response in natural language.

The Ontology bot requires that the NLU state contains three annotations to get activated:

1. Identified entities linked with Wikidata by the Entity Linker
2. Topic annotation of the user utterance
3. Intent annotation of the user utterance

Currently, the ontology bot supports the following topics: movies, books, music, sports and video games. An additional restriction is represented by the intent. We trigger the linking concept procedure only when the intent is one of the following: *tell\_me\_about* or *user\_preference*. For the latter, we support two different modalities that are triggered according to the type of preference expressed by the user. If the user expresses a positive preference towards the mentioned entity (e.g. "I love Ryan Gosling"), we trigger the linking concept procedure described before. Otherwise, we rely on the semantic information associated to the entity (such as "occupation" or "type") in order to ask to the user if they want to talk about something else (e.g. "Oh I'm sorry that you don't like Ryan Gosling. Is there another *actor* that you enjoyed?"). We argue that this architecture has two main advantages: 1) the strict restrictions in terms of NLU annotations enforce the bot to remain on-topic while generating domain-specific responses conditioned on the user intent as well as identified entities; 2) additional domains can be easily added by creating a new response generator specific for that domain.

The Ontology bot has been designed with *Linked Data* (Bizer et al., 2011) principles in mind. In particular, by linking specific entity mentions to specific entities in the Wikidata knowledge base we are not limited to the information stored in it. On the other hand, the Wikidata knowledge base allows us to find representations of a specific entity in other knowledge bases or other websites. An interesting use case of this feature is represented by "trivia" coming from *IMDb*. Specifically, once an entity has been recognised, we check whether a link to IMDb is associated to it in Wikidata and we retrieve additional information associated to the entity that goes behind simple factual knowledge stored in Wikidata.

While the enforced restrictions on the ontology bot directly encourage the conversation to be topic and intent specific, on the other hand it makes the bot highly dependent on the NLU pipeline. In case of NLU overload/failure, the ontology bot would specifically remain unresponsive. We addressed this issue by monitoring and fail-safe mechanisms (see Section 3.1.6).

### 3.2.2 Abuse Mitigation Bot

During the 2017 competition, we encountered many interactions containing some form of profanity or abuse towards the system (we estimate this to be around 5%). In order to deal with such requests, we trained an abuse-detection model and designed response strategies aimed at mitigating such behaviour. Note that we differentiate between the use of foul language ("*He's an idiot*") and abuse ("*You are an idiot*") and do not aim to mitigate the former.

We use a subset of ~82K user utterances collected during the 2017 competition. Of these, we manually annotated 4050 user turns, which were pre-filtered using a keyword-based detector. The rest of the utterances were automatically labelled as non-abusive. Of the manually annotated utterances, 1342 were offensive and 822 were sexual/hate speech.

We trained an embedding-based abuse detection model using RASA-NLU<sup>8</sup>. The model distinguishes between non-abusive, generally offensive utterances, and sexually-charged requests or hate-speech. Table 1 shows the precision, recall and F1-score of the model on a test set of 8242 of utterances. In the case of profanity detection, we wish to emphasize precision over recall as false positives can be confusing to the user and prompt them to end the conversation. Note that recall for the non-abusive class is 1.0.

---

<sup>8</sup><https://rasa.com/docs/nlu/>

	Precision	Recall	F1-score
Non-abusive	0.98	1.00	0.99
Offensive	0.84	0.18	0.29
Sexual/Hate	1.00	0.05	0.09

Table 1: Abuse bot performance statistics.

When a given user utterance is labelled as offensive, Alana will produce a response designed to discourage such abuse. We have currently implemented several strategies based on existing research on bullying mitigation which include: changing topic, appealing to authority (“Would you like me to send a transcript of this conversation to you contact: Mum?”), chastising the user (“Do you talk to your mother like that?”), among others. These strategies are outlined in more detail in Cercas Curry & Rieser (2018). In future work, we will study the suitability of these mitigation strategies in the context of social dialogue between a human and a bot.

### 3.2.3 Reddit bot

In order to make the system more engaging and conversational, we source humorous comments from Reddit. We scraped popular subreddits such as “ShowerThoughts” and “Today I Learned” and indexed them using Lucene. At runtime, we search the index for any noun phrases mentioned in the user’s utterance. Below are some example utterances - see also Fig. 4:

**User:** I like pizza.

**Reddit\_bot:** whenever a pizza commercial has stretchy melty cheese, it looks good and appealing, but when it happens in real life it’s annoying and messy.

**Reddit\_bot:** a pizza delivery sign on top of a 2017 mustang is the perfect example of misplaced priorities.

**User:** I like sharks

**Reddit\_bot:** martin luther king jr. only gets a day, but sharks get a week.

**User:** Tell me about cats

**Reddit\_bot:** the only time cats display true happiness on their face is when they are sleeping

A comparison of the average scores of conversations containing Reddit vs. those without Reddit utterances shows that the former receive higher scores (3.52, based on 200 dialogues, compared to 3.21, based on 147 dialogues from the same day, which did not include Reddit responses).

### 3.3 Updated bots from Alana v1

In addition to the newly introduced bots described above, the current Alana system also includes updated versions of bots developed during last year’s competition (Papaioannou et al., 2017b):

**Coherence bot** This bot is responsible for keeping the flow of the conversation as natural and coherent as possible. It is doing so by initially building a very basic user model based on the user’s preferences during the introduction of the conversation, which can later on utilise when trying to talk about something different (would prefer talking about something that the user already said he enjoys rather than something that the user clearly disliked). This is the only bot in the ensemble that utilises the system’s mixed initiative capability, being able to switch the topic if the conversation demands it. It generates a response *driver* – a question or statement on the current or switched topic on *every* turn, which can then be used by any bot’s response that was picked by the *selection strategy* during the *post-processing* stage (see section 2).

The bot uses a priority strategy when generating a driver:

1. Current topic (tries to stay on topic)
2. User’s preference (picks a topic that the user enjoyed in the past)

3. Rapport building question (asking the user a question in an attempt to enrich the user model)
4. Generic driver on different topics

The *coherence bot* is also able to use a previously built user’s model in case of a returning user, like in the example dialogue in fig. 4.

**Wiki bot** generates responses based on a related Wikipedia article to an entity mention in the user utterance. In more detail, we first created an index of the English portion of Wikipedia, by keeping the title of each article and the first sentence of each section, using Lucene. Then we query it by *incrementally* (i.e., if one strategy fails to return any results, we progress to the next one) including as search terms: a) all named entities recognised by the NLU pipeline, b) all noun phrases that appear in the user query, c) all 1,2,3-ngrams of the user utterance after removing stop words and applying lemmatisation. Next, we keep the top-5 retrieved articles, and deliver the first sentence of each section piecemeal. In this way, we avoid creating a very long system response, and the user controls the amount of information presented to them over multiple turns.

**News bot** creates responses based on multi-sentence summaries from news articles that contain a named entity mentioned by the user. In particular, we crawl several online news sites containing thousands of news articles using News API<sup>9</sup> every night and update an index powered by Lucene. For every retrieved article, we keep a 4-sentence summary created using LexRank (Erkan & Radev, 2004), a popular extractive summarizer<sup>10</sup>. Then for each named entity recognized by the NLU pipeline in the user utterance, we retrieve the up to 5 most recent news summaries. Similarly to Wiki bot, we present them sentence-by-sentence, via a simple interaction with the user across multiple turns.

**Persona** is an AIML-based handcrafted bot that responds to specific user questions, mainly those related to Alana’s personality. It also handles requests regarding sensitive topics, such as suicide. Persona’s patterns have been expanded and refined to better capture user intents.

**Evi** is a simple wrapper around the Amazon Evi question answering engine.<sup>11</sup> Since last year, we have allowed Evi to lock on factual questions such as “Who is Donald Trump?”.

**Facts/Jokes/Stories bot** responds to user requests such as “tell me a joke” or “tell me a fun fact”. It uses a hand-collected database of fun facts on several topics (e.g., nature, history, geography), jokes, and short stories.

**Eliza** is an extension of the simple, Eliza-style AIML-based chatbot Rosie.<sup>12</sup> Eliza’s replies are mostly followed by questions from the Coherence bot. We edited the Eliza bot to remove some responses which sounded overly rude or challenging to the user.

## 4 An Example Dialogue

An example dialogue with Alana is shown in Figure 4. This is a real example of Alana conversing with one of the developers, rather than being real user data, and it illustrates a variety of the conversational AI features implemented in the system:

- Turn 5 shows the fragment/ellipsis resolution performed by the NLU (the bot ensemble receives the full sentence).
- Turns 5, 9 and 11 show responses by the Coherence bot (see Section 3.3). The bot tries to track which topics the user likes (Turns 5 and 11) and stay on the current topic (Turn 11), but it also accepts user’s requests to change the topic (Turn 9). Coherence bot prompts are also appended to other bots’ responses (Turns 6, 7, 8, 10, 14).

---

<sup>9</sup><https://newsapi.org/>

<sup>10</sup>We used the Python implementation sumy found here: <https://github.com/miso-belica/sumy>

<sup>11</sup><https://www.evi.com/>

<sup>12</sup><https://github.com/pandorabots/rosie>

- Turns 6, 10, 12 and 14 feature responses by the Ontology bot for different topic-specific ontologies (movies, music, and books). The movies response in Turn 6 includes trivia obtained from IMDb (see Section 3.2.1).
- Turn 7 shows a generic reply to a generic user input, handled by the Persona bot (see Section 3.3).
- Turn 8 shows co-reference resolution performed by the NLU and an entity explanation performed by Ontology bot (see Section 3.3).
- Turn 13 shows the Clarification service prompting the user to disambiguate a named entity (see Section 3.1.3).
- Turn 15 includes an example of a response by the recently added Reddit bot (see Section 3.2.3).

## 5 System Evaluation

Below we describe *A/B tests* performed during this year’s competition investigating the effect on the performance of the system of removing/modifying certain components of the system. As a baseline, we consider our average score at the end of last year’s competition which is based on user-rated 30,269 dialogues. The scores range from 1-5, 5 being the highest, and are given by users after chatting with our system.

### 5.1 AIML-bot Ablation Study

As a first test, we consider the effect of our AIML bots, which produce rule-based responses as described in Sec. 3.3. Each bot was disabled consecutively for approximately one week.

Bot Disabled	Score	Number of Dialogues
none (=Baseline)	3.56	30,269
Persona	3.51	5,632
ELIZA	3.45	6,388
Both	3.41	5,301

Table 2: Average scores of the Alana system with and without its rule-based components.

Table 5.1 shows the average user rating during the study. The removal of Persona has a modest effect on the average score. We hypothesise that this is due to other bots, like Evi and Eliza, being able to answer to many of the requests that are generally handled by Persona.

In contrast, ELIZA’s removal has a strong effect on the scores, which were reduced by 0.11. Unlike our other bots, ELIZA is designed to be a stand-alone bot and is so able to handle generic user utterances, such as “I don’t know”, as well as Persona-like utterances. As such, removing Eliza also removes Alana’s fail-safe. In addition, Eliza’s responses are inquisitive in nature and tend to prompt further responses from the user.

Finally, removing both Persona and ELIZA results in an even lower score. This is expected and suggests that ELIZA plays a big role in keeping the scores relatively high in the case where Persona is removed.

### 5.2 Introducing Ontologies

After the complete revamping of the Alana architecture we decided to introduce a preliminary version of the Ontology bot on 13th June 2018. In this first release, the bot was able to support only the movie topic by providing linking concepts for different entity types such as movies, actors, producers and directors. In addition, it was able to provide Trivia for the entity mentioned by the user, trivia retrieved from IMDb. In order to assess the effectiveness of this bot, we analysed our conversations since the 1st July 2018 to 7th July 2018 by comparing the average score of the dialogues containing ontology bot responses and of the average score of the dialogues not containing them. Figure 5 shows the average scores of the collected dialogues. We can definitely spot a positive trend starting from 1st

		[after four greeting turns...]	
5	<b>Alana:</b>	Welcome back then USERNAME. Since I remember that you like movies, I am looking forward to the new movie Jurassic world that was just released. What movie are you looking forward to watching?	
	<b>User:</b>	Vanilla Sky [→ “I’m looking forward to watching Vanilla Sky ”]	Fragment resolution
6	<b>Alana:</b>	Oh I heard something interesting about Vanilla Sky. In the cell, Carl Jung’s book, Memories, Dreams, and Reflections can be seen on the table between David and McCabe. The book is all about Jung’s personal dreams and how they helped him uncover his shadow and remove his persona or mask. Speaking of movies. What actors do you like? I like Leonardo dicaprio a lot. He was awesome in the Revenant.	Movie Trivia
	<b>User:</b>	Interesting	
7	<b>Alana:</b>	I just think the world is full of strange and unexpected things like that. So, talking about movies, What famous actor or actress would you like to meet? I would love to meet Will Smith. He’s just so funny!	Persona + Coherence
	<b>User:</b>	who is he?	Co-reference Resolution
8	<b>Alana:</b>	I guess you know that Will Smith is American actor, film producer and rapper. his best movies are Men in Black II, Enemy of the State, Men in Black 3, and Men in Black and he has won the Grammy Award for Best Rap Solo Performance. So USERNAME, on the topic of movies. Whats a movie that you like? I love Star Wars the Force Awakens. I think Chewbacca is my favourite character.	Movie Ontology: Entity explanation
	<b>User:</b>	I want to talk about music	Topic navigation
9	<b>Alana:</b>	So, talking about music. Who is a singer that you like? I really like Taylor Swift, because her voice is so beautiful.	Coherence
	<b>User:</b>	Nina Simone [→ “I like Nina Simone ”]	Fragment resolution
10	<b>Alana:</b>	Yes. Someone was talking about Nina Simone the other day. Apparently, Daughter Lisa Celeste ‘Lisa Simone Kelly’ , born in 1962. She’s a Broadway actress, star of the Broadway musical Elton John’s Aida . Oh, okay. What is a musician that you like? I love Beyonce.	Music Ontology
	<b>User:</b>	could we talk about something else?	Topic change request
11	<b>Alana:</b>	So USERNAME, since I know you like books , what have you read recently? I am reading Diary of a wimpy kid.	User modelling
	<b>User:</b>	Handmaid’s Tale [→ “I read Handmaid’s Tail ”]	Fragment resolution
12	<b>Alana:</b>	Oh so you like to read a good science fiction? Maybe you know that the book Do Androids Dream of Electric Sheep? has the same genre. You might like that if you haven’t already read it. What’s another book like The Handmaid’s Tale that you enjoyed?	Book ontology: recommendation
	<b>User:</b>	Let’s talk about Tom Hardy	
13	<b>Alana:</b>	Oh I know tom hardy. Do you mean Tom Hardy the English actor, screenwriter and producer or Thomas Hardy the English novelist and poet?	Clarification interaction
	<b>User:</b>	I mean the novelist	
14	<b>Alana:</b>	Yes. Some friends told me something interesting about Thomas Hardy. He was an architect before becoming a full-time writer. So USERNAME, on the topic of movies. Who is a movie hero that you like to watch? I like Han Solo.	Book ontology: trivia
	<b>User:</b>	What do you know about pizza?	
15	<b>Alana:</b>	I was thinking that whenever a pizza commercial has stretchy melty cheese, it looks good and appealing, but when it happens in real life it’s annoying and messy.	Reddit response retrieval

Figure 4: Example Alana interaction with feature annotations (see Section 4 for details).

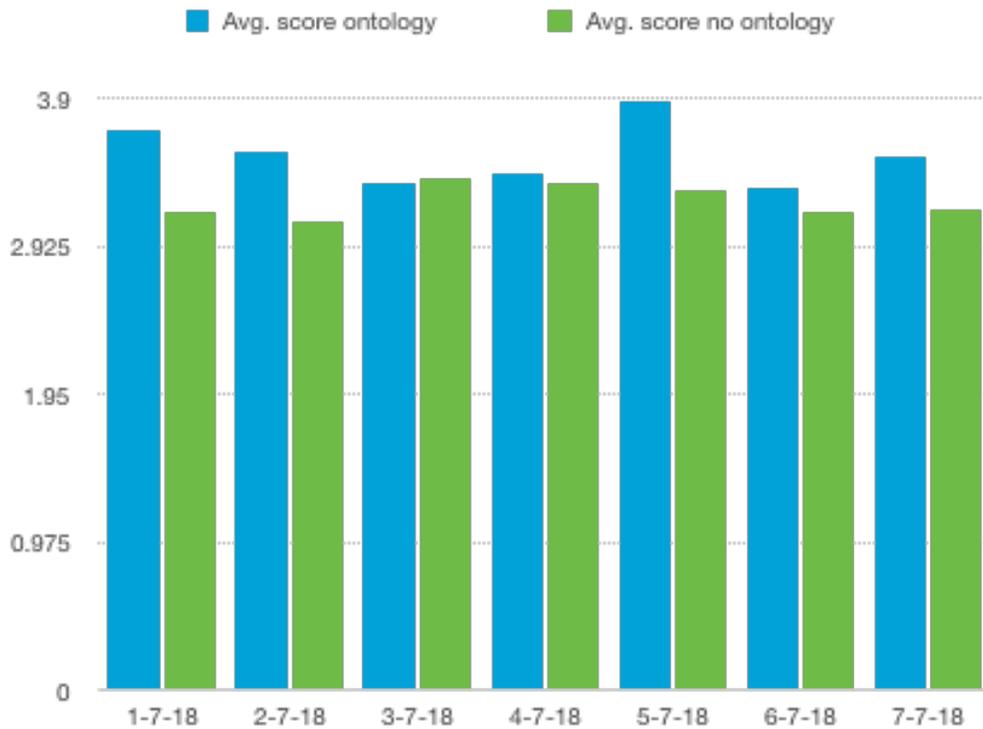


Figure 5: Average score comparison between conversations containing responses produced by the Ontology bot (reported in blue) and conversations without any ontology responses (reported in green) from 1st July 2018 to 7th July 2018.

July 2018 were the average scores for the collected dialogue are always better than the ones without ontologies responses. A Mann-Whitney-U Test confirms the improvement in scores is statistically significant ( $p < 0.01$ ). After this preliminary evaluation, we extended the movie ontology bot by introducing support for fictional characters (e.g. Han Solo) and media franchises (e.g. Star Wars). We further released a music ontology on July 4th 2018 by supporting linking concepts and trivia for bands and artists. Almost in the same release, we deployed a book ontology able to provide linking concepts for writers, books and fictional characters.

Figure 6 depicts the average score comparison for dialogues with and without the ontology bot responses from 7th August 2018 to 13th August 2018. From the average scores it is possible to appreciate the benefit of having a bot capable of dealing with entities that belong to different domains. In particular, in the final release of the Ontology bot we were able to generate responses for the following topics: movies, books, music and video games. The integration of this bot brought a substantial improvement of the performance, with an average improvement of 0.32 points, which suggests the Ontology bot is well-suited to the strategy of prompting the user to talk about entities that they like.

### 5.3 Response ranking: engagingness experiments and evaluation

With the main challenge’s objective defined as ‘long and engaging’ conversations, we attempted to find out which dialogue properties reflect this behaviour (Shalyminov et al., 2018).

For that, we performed a correlation analysis of user ratings we have been receiving over the 2017 challenge period and the aspects of the dialogue directly reflecting the objective: dialogue length and explicit user feedback/engagement. We used the ratio of users’ turns containing some predefined key phrases such as “that’s pretty cool”, “you’re funny”, “gee thanks” or “awful”, “you’re dumb” together with the sentiment polarity of those turns as the approximation of user feedback and engagement (in



Figure 6: Average score comparison between conversations containing responses produced by the Ontology bot (reported in blue) and conversations without any ontology responses (reported in green) from 7th August 2018 to 13th August 2018.

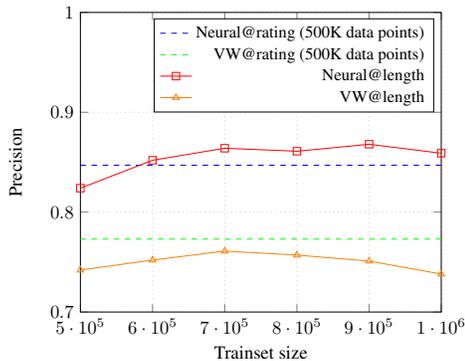


Figure 7: Comparison of rankers trained on extended datasets

Variables	Pearson corr. coefficient
rating/length	0.11
rating/positive feedback	0.11
rating/negative feedback	0.04
length/positive feedback	<b>0.67</b>
length/negative feedback	0.49

Table 3: Correlation study of key dialogue aspects

total, we collected about 600 such utterances). As shown in Table 3, the correlation study wasn't able to show any relationship between ratings and the dialogue properties of interest. On the other hand, dialogue length itself has a promising moderate correlation with users' positive feedback (although the length/negative feedback correlation is also present, it's slightly weaker).

This insight led us to the following experiment with the bot ensemble's response ranker (as was originally used in Papaioannou et al. (2017a)). We compared two alternative training signals for the ranking models: dialogue length and rating. The ranking function works as follows: given dialogue context (3 last turns) and the response candidate, the ranking model outputs a point-wise score for this pair in the range [0, 1], so that the candidate with the highest score gets selected for the given context. For training, we employed the following approximation: each turn in a dialogue is assigned

the same target score — length or rating of the full dialogue, normalized at  $[0, 1]$ . The intuition behind that is, given a sufficient amount of data, consistently good context-response pairs will emerge from the successful dialogues, and less certain cases will be naturally filtered out.

For training, we collected several datasets of context-response pairs and corresponding target scores (lengths and ratings). The ‘Rating’ dataset totals 500,000 data points (at 8:1:1 split for training, development, and testing) — roughly limited by the number of rated turns we ended up having after data preprocessing. The ‘Length’ datasets span from 500,000 to 1,000,000 data points. All those datasets consist of positive examples (target score  $> 0.7$ ) and negative ones (target score  $< 0.3$ ) with equal proportion<sup>13</sup>.

For evaluation, we collected an independent user feedback-based dataset: it consists of *context/gold response/fake response* tuples where gold responses are those system turns followed by explicit positive user feedback (determined as described above), and fake ones are random system turns from the corpus. We measured the models’ precision at ranking gold responses higher than the corresponding fake ones: compared are the Neural and VowpalWabbit (VW) rankers (Shalyminov et al., 2018) trained from either length or rating signal. As shown in Figure 7, apart from the case of our smallest dataset (500,000 examples), the length-based neural ranker consistently outperforms the rating-based one given more unlabelled training data available. The VW’s performance doesn’t reflect this trend — the reason for that might be the model’s inherent lower capacity.

Overall, this study showed the following: firstly, user ratings aren’t stable enough to be an optimisation objective, in our case for a response ranking model. Secondly, conversation length itself can be used as one for generating long and engaging social dialogues — i.e., we demonstrated that given a sufficient amount of raw dialogue transcripts, we were able to train models of a similar and superior quality than ones trained on user ratings.

Looking ahead, this finding can potentially make data collection for social conversational agents simpler and less expensive in the future<sup>14</sup>.

## 6 Overall Performance

We achieved consistently high user ratings and long conversations throughout the semi-finals period. Our final average user score over the whole challenge was 3.4, with an average duration of 2.19 and 11.3 turns. 10 percent of conversations were over 9 minutes long. Our uptime for the period was 99%.

Figure 8 shows our daily average scores over the semi-finals period. Several datapoints are noteworthy:

- Days 7 and 42, 43 were high-traffic days after the “Let’s Chat” skill was advertised in an email newsletter. We note that our system was more robust to this later in the competition (days 42 and 43).
- No scores are available for days 9 and 10 – we did not get any data for these dates due to revamping our systems to add security measures to prevent intrusions.
- On Day 19 our NLU pipeline was down due to timeouts on the Neptune cluster (later worked-around using the fail-safe mechanism described in Section 3.1.6). This can be seen as a lower bound on the performance of our system.
- Day 45: we added the Reddit bot and observed a performance boost.
- We note a particularly strong period over days 17-25, with 4 scores over 3.5 in that period. Further analysis will be needed to establish reasons for this, since the system was debugged further from this point onwards, which should have only led to increased performance.
- We note a period from day 30 to 34 where our performance was decreasing, in contrast to the overall upward trend. We believe that there are probably 2 reasons for this, all of which lead to reduced number of Ontology bot responses:
  1. Timeouts in the Ontology bot caused by slow performance of the IMDb API used to obtain trivia (see Section 3.2.1). We later solved this by caching.

<sup>13</sup>Using more extreme thresholds did not produce enough data while less ones did not provide adequate training signal

<sup>14</sup>While this model was not yet deployed for the semi-finals, due to the presence of multiple new bots and their outputs, we plan to retrain it and experiment with it in future, based on data collected during that period.

- Timeouts on the Neptune cluster, which were later worked-around using an auto-restart mechanism (see Section 3.1.6).

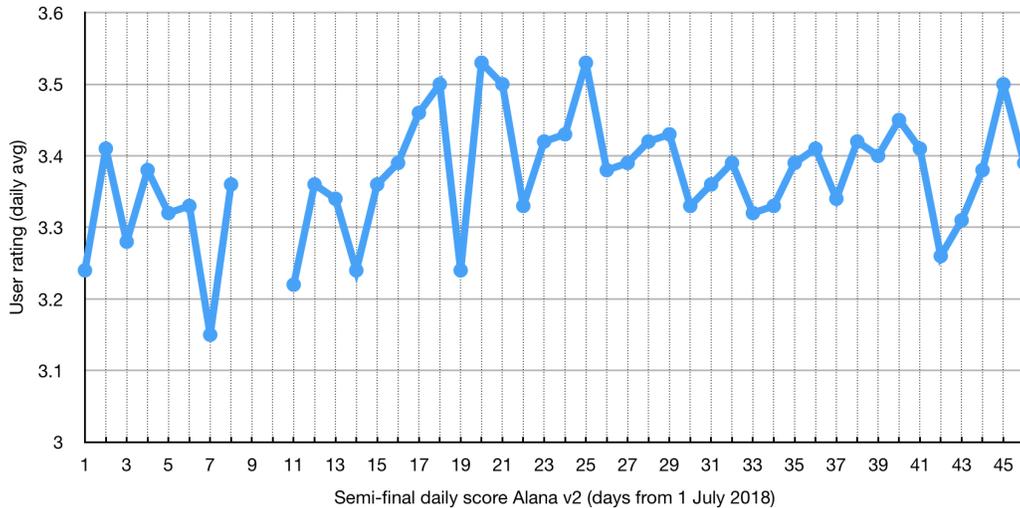


Figure 8: Semi-final results: Heriot-Watt Alana v2 system (notes: no scores for days 9 and 10 – no data as we revamped our systems to add security measures to prevent intrusions. Days 7 and 42, 43 were high-traffic days after advertising the “Let’s Chat” skill in a newsletter. On Day 19 our NLU pipeline was down. Day 45 – added Reddit bot.

## 7 Future Enhancements

We plan a number of improvements to Alana, some of which are new features, and some enhancements of existing ones:

- Improvements to the NLU pipeline:
  - Better entity linking using an ensemble of FEL and NECKAr<sup>15</sup>
  - Better clarification functionality using the topic of the conversation to filter out irrelevant entities
  - Improved coreference resolution
  - Handling Multiple Intents in NLU: at the moment, our NLU pipeline can only handle a single intent/dialogue act per turn, and this often leads to inappropriate responses.
  - Better ASR error handling
- Multi-turn ontologies + Question Answering: currently Alana’s interactions about particular NEs are shorter than we would like. We plan to extend the Ontology-bot to handle longer stretches of conversation about a particular NE, or set of related NEs, including the ability to answer user questions about these entities.
- Extension of ontology bot to different domains: currently we are in the process of deploying sports ontology live after the semi-finals
- Better Persona responses with opinions about entities (Why do you like Roger Federer?)
- Improving recall of our abuse detection model and testing different mitigation strategies
- Improving the precision of the search engine used by Reddit-bot

<sup>15</sup>[https://event.ifi.uni-heidelberg.de/?page\\_id=532](https://event.ifi.uni-heidelberg.de/?page_id=532)

## 8 Conclusion

We have described our 2018 Alexa prize system (called ‘Alana’) – an ensemble of bots. We detailed the main advances over our 2017 Alana system: (1) a deeper Natural Language Understanding (NLU) pipeline; (2) the use of topic ontologies and Named Entity Linking to enable the user to navigate and search through a web of related information; (3) system generated *clarification questions* to interactively disambiguate between Named Entities; (4) a profanity & abuse detection model; and (5) response retrieval from Reddit. We also presented results from several ablation studies that measure the performance contributions of specific features (e.g. use of Ontology-bot, Reddit-bot, etc). We find that each component of Alana measurably improves the average user score by increasing engagement and coherence. In particular, we find an important contribution from ontologies to improving the quality of the conversation, with an increase of around 0.3 points on the Likert scale. Our final score, namely averaged user ratings over the whole semi-finals period, was 3.4.

### Acknowledgments

Many thanks to Jeff Nunn for his excellent technical support of the project. We also thank Sanju Pancholi, for fruitful weekly discussion about Alana’s performance.

### References

- Bizer, Christian, Heath, Tom, and Berners-Lee, Tim. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pp. 205–227. IGI Global, 2011.
- Blanco, Roi, Ottaviano, Giuseppe, and Meij, Edgar. Fast and space-efficient entity linking in queries. In *Proceedings of the Eight ACM International Conference on Web Search and Data Mining, WSDM 15*, New York, NY, USA, 2015. ACM.
- Cercas Curry, Amanda and Rieser, Verena. #metoo: How conversational systems respond to sexual harassment. In *Proceedings of the Second ACL Workshop on Ethics in Natural Language Processing*, pp. 7–14. Association for Computational Linguistics, 2018.
- Chen, Danqi and Manning, Christopher. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, 2014.
- Erkan, Günes and Radev, Dragomir R. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004. ISSN 1076-9757. URL <http://dl.acm.org/citation.cfm?id=1622487.1622501>.
- Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>.
- Papaioannou, Ioannis, Cercas Curry, A, Part, Jose L, Shalymov, Igor, Xu, Xinnuo, Yu, Yanchao, Dušek, Ondrej, Rieser, Verena, and Lemon, Oliver. Alana: Social dialogue using an ensemble model and a ranker trained on user feedback. *Proc. AWS re: INVENT*, 2017a.
- Papaioannou, Ioannis, Dondrup, Christian, Novikova, Jekaterina, and Lemon, Oliver. Hybrid chat and task dialogue for more engaging hri using reinforcement learning. In *Proc. RO-MAN. IEEE*, 2017b.
- Pappu, Aasish, Blanco, Roi, Mehdad, Yashar, Stent, Amanda, and Thadani, Kapil. Lightweight multilingual entity extraction and linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 17*. ACM, 2017.
- Pennington, Jeffrey, Socher, Richard, and Manning, Christopher D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL <http://www.aclweb.org/anthology/D14-1162>.
- Shalymov, Igor, Dušek, Ondřej, and Lemon, Oliver. Neural response ranking for social conversation: A data-efficient approach. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*. Association for Computational Linguistics, 2018.
- Shen, Wei, Wang, Jianyong, and Han, Jiawei. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015.
- Straková, Jana, Straka, Milan, and Hajič, Jan. Open-source tools for morphology, lemmatization, pos tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 13–18, 2014.