Using subtitles to deal with Out-of-Domain interactions

Daniel Magarreiro, Luísa Coheur, Francisco S. Melo
INESC-ID / Instituto Superior Técnico
Universidade Técnica de Lisboa
Lisbon, Portugal
name.surname@tecnico.ulisboa.pt

Abstract

This paper explores the possibility of using interactions between humans to obtain appropriate responses to Out-of-Domain (OOD) interactions, taking into consideration several measures, including lexical similarities between the given interaction and the responses. We depart from interactions obtained from movie subtitles, which can be seen as sequences of turns uttered between humans, and create a corpus of turns that can be used to answer OOD interactions. Then, we address the problem of choosing an appropriate answer from a set of candidate answers, combining several possible measures, and illustrate the results of our approach in a simple proof-of-concept chatbot that is able deal with OOD interactions. Results show that 61.67% of the answers returned were considered plausible.

1 Introduction

Recent years have witnessed the appearance of *virtual assistants* as a ubiquitous reality. Well-known examples include *Siri*, from Apple, *Anna*, from IKEA, and the buttler *Edgar Smith*, at Monserrate Palace (see Fig. 1).

Such systems are typically designed to interact with human users in well-defined domains, for example by answering questions about a specific subject or performing some pre-determined task. Nevertheless, users often insist in confronting such domain-specialized virtual assistants with OOD inputs.

Although it might be argued that, in light of their assistive nature, such systems should be focused in their domain-specific functions, the fact is that people become more engaged with these applications if OOD requests are addressed (Bickmore and Cassell, 2000; Patel et al., 2006).

Figure 1: The virtual buttler, Edgar Smith, which can be found at Monserrate Palace, in Sintra, Portugal (Fialho et al., 2013).

Current approaches are able to address specific OOD interactions by having the system designer handcraft appropriate answers. However, it is unlikely that system designers will be able to successfully anticipate all the possible OOD requests that can be submitted to such agents. An alternative solution to deal with OOD requests is to explore the (semi-)automatic creation/enrichment of the knowledge base of virtual assistants/chatbots, taking advantage of the vast amount of dialogues available at the web. Examples of such dialogues include those in play/movie scripts, already used in some existing systems (Banchs and Li, 2012).

In this paper, we follow (Ameixa et al., 2014) and adopt an alternative source of dialogues, namely *movie subtitles*. The use of movie subtitles brings two main advantages over scripts and other similar resources. First, the web offers a vast number of repositories with a comprehensive archive of subtitle files. The existence of such collection of subtitle files allows data *redundancy*, which can be of great help when selecting the adequate reply to a given OOD request. Secondly, subtitles are often available in *multiple languages*, potentially

enabling multilingual interactions.[1]

Our approach can be broken down into two main steps, representing our contributions. First, we describe the process of building an improved version of *Subtle*, a corpus of interactions, created from a dataset of movie subtitles. Secondly, we describe a set of techniques that enables the selection/retrieval of an adequate response to a user input from the corpus. The proposed techniques are deployed in a dialogue engine, the *Say Something Smart* (SSS), and an evaluation is conducted illustrating the potential behind the proposed approach in addressing OOD interactions.

This paper is organised as follows. Section 2 surveys some related work. Section 3 describes the construction of the *Subtle* corpus. The SSS engine is described in Section 4 and Section 5 presents the results of a preliminary evaluation. Section 6 concludes, pointing directions for future work.

2 Related work

Virtual assistants have been widely used to animate museums all over the world. Examples include the *3D Hans Christian Andersen* (HCA), which is capable of establishing multi-modal conversations about the namesake writer's life and tales (Bernsen and Dybkjaer, 2005), *Max*, a virtual character employed as guide in the Heinz Nixdorf MuseumsForum (Pfeiffer et al., 2011), the twins *Ada* and *Grace*, virtual guides in the Boston Museum of Science (Traum et al., 2012) and *Edgar Smith* (Fialho et al., 2013), a virtual butler that answers questions about the palace of Monserrate, in Sintra, Portugal (see Fig. 1).

However, and despite the sophisticated technology supporting these (and similar) systems, they are seldom able to properly reply to interactions that fall outside of their domain of "expertise"[2], even though such interactions are reported as quite frequent. For instance, Traum et al. (Traum et al., 2012) report that 20% of the interactions with *Ada* and *Grace* are inappropriate questions.

In order to cope with such OOD interactions, several approaches have been proposed in the literature. For example, when unable to understand a specific utterance (and formulate an adequate answer), *Edgar* (Fialho et al., 2013) suggests questions to the user. In the event that it is repeatedly unable to understand the user, *Edgar* starts talking about the palace. Finally, in order to mitigate the effect of such misunderstandings on the user's engagement and perception of agency, *Edgar* was designed to "blame" his age and bad hearing for its inability to understand the user. In a different approach, *HCA* (Bernsen and Dybkjaer, 2005) changes topic when lost in the conversation. Also, much like *Edgar*, *HCA* has been designed with an "excuse" for not answering some questions: the "virtual HCA" does not yet remember everything that the "real Hans Christian Andersen" once knew. *Max* (Pfeiffer et al., 2011) consults a web-based weather forecast when queried about the weather, and Wikipedia, when approached with factoid questions (Waltinger et al., 2011). In (Henderson et al., 2012), a set of strategies to deal with non understandings is proposed.

Recently, Banchs and Li introduced *IRIS* (Banchs and Li, 2012), a chat-oriented dialogue system that includes in its knowledge sources the *MovieDiC* corpus (Banchs, 2012) . The *MovieDiC* corpus consists of a set of interactions extracted from movie scripts that provides a rich set of interactions from which the system can select a plausible reply to the user's input.

In this paper we take this idea one step further, and propose the use of movie subtitles to build a corpus for open-ended interactions with human users. Subtitles are a resource that is easy to find and that is available in almost every language. In addition, as large amounts of subtitles can be found, linguistic variability can be covered and redundancy can be taken into consideration (if a turn is repeatedly answered in the same way, that answer is probably a plausible answer to that turn).

3 From subtitles to interactions: Building the *Subtle* corpus

In this paper we use knowledge bases constituted of *interactions*, an approach already used in other existing systems (Traum et al., 2012). Each interaction (adjacent pair) comprises two turns, (T, A), where A corresponds to an answer to T, the *trigger*.[3] The following are examples of interactions:

[1] In this paper, we will focus on English, although some experiments with Portuguese were also conducted.

[2] Check `http://alicebot.blogspot.pt/2013/07/turing-test-no-sirie.html` to see *Siri* (Apple's virtual assistant) answers to the 20 questions of the 2013 Loebner Prize contest.

[3] We use the word *trigger*, instead of the usual designation of *question*, since not every turn includes an actual question. Throughout the text, we also use the designations *input* and

```
(T1: You know, I didn't catch your age.
     How old are you?,
 A1: 20)
```

```
(T2: So how old are you?,
 A2: That's none of your business)
```

In this section we describe the process of building interaction pairs based on movie subtitles. We designed a configurable process for building the corpus that takes into consideration the language of the subtitles being processed (henceforth, English and Portuguese) and other elements that should be considered when building the corpus, such as the time elapsed between two consecutive subtitles. Independently of the particular configuration adopted, we refer to the corpus thus built as *Subtle*, although different configurations will evidently lead to different corpora. This corpus is an improved version of the one described in (Ameixa and Coheur, 2014) and (Ameixa et al., 2014).

3.1 Subtitles: The starting point

We obtained 2Gb of subtitles in Portuguese and English from *OpenSubtitles*.[4] These files are in the `srt` format, which consists of a sequence of slots, each containing the following information:

1. The *position* of the slot in the sequence.
2. The *time* indicating when the slot should appear/disappear on the screen.
3. The *content* of the subtitle.

A blank line indicates the start of a new slot. An example of a snippet from a subtitle's file is depicted in Fig. 2.

The 2Gb of subtitle data used includes many duplicate movie subtitles that were removed. In particular, we obtained a total of $29,478$ English subtitle files corresponding to a total of $5,764$ different movies. In removing the duplicate entries, we selected the subtitle file containing the largest number of characters. Similarly, we obtained a total of $14,679$ Portuguese subtitle files corresponding to a total of $3,701$ different movies. In the end, the *Subtle* corpus was built from $5,764$ English subtitle files and $3,701$ Portuguese subtitle files.

request to refer to user turns.

[4] `http://www.opensubtitles.org/`

```
770
01:01:05,537 --> 01:01:08,905
And makes an offer so ridiculous,

771
01:01:09,082 --> 01:01:11,881
the farmer is forced to say yes.

772
01:01:12,752 --> 01:01:15,494
We gonna offer to buy Candyland?
```

Figure 2: Snippet of a subtitle file.

3.2 Extracting interactions from subtitles

We now describe the process of extracting interactions from the selected subtitles files.

Cleaning data

Besides the actual subtitles, there is information provided in the subtitle files that is irrelevant for dialogue and should, therefore, be removed. Examples of portions removed include those containing:

Characters' names. Some subtitle files include the name of the speaker at the beginning of the utterance (e.g., *Johnny: Oh hi, Mark.*). This is particularly useful both when a character is not appearing on the screen and for hearing impaired watchers. Since such names should not be included in the responses of our system, they were eliminated in every turn they appear.

Sound descriptions for hearing impaired. It is also common for subtitle files to include the sound descriptions being played that are relevant for the watcher to perceive (e.g. `[TIRES SCREECHING]`). Such descriptions are not actual responses, so we removed them from the corpus.

Font-changing tags. Subtitles sometimes include tags that video players can interpret to change the normal font in which the tagged subtitle is to be displayed (e.g. `<font color="#ffff00" size=14> Sync by honeybunny </font>`). Such tagged subtitles seldom contained any dialogue element and, therefore, were eliminated.

Finding real turns

The main challenge in building the *Subtle* corpus is to decide which pairs of consecutive slots in the subtitle file correspond to an actual dialogue and which ones do not (and instead correspond, for instance, to a scene change).

In contrast to the version of *Subtle* described in (Ameixa et al., 2014), we allow the user to configure the maximum time allowed between two slots for them to be considered part of a dialogue and used to build an interaction pair. For example, if that time is set to 1 second and two slots are separated by more than that period, they will not be considered as an interaction pair. However, a hard time threshold is difficult to set appropriately, and may lead to useful interactions being discarded from the corpus, if the corresponding value is not adequately set.

To mitigate the impact of a hard time threshold, we also allow the possibility of setting the value of the maximum time between slots to 0, in which case *all* consecutive pairs of slots are considered to be part of a dialogue and used to construct an interaction pair. This latter option ensures that the corpus will contain all the information in the subtitles, but also means that many interaction pairs that are not real interaction pairs in a dialogue will be present in the corpus. As will soon become apparent, we compensate for this disadvantage by including a "soft threshold" mechanism when choosing an answer from a set of possible answers.

Another challenge in processing the subtitles stems from the fact that there is not a standard formatting followed by all the subtitle creators. To handle these formatting differences, we identified common formatting patterns in the process of building the *Subtle* corpus, and specialised, handcrafted rules were designed to take care of such patterns. For instance, when two consecutive subtitle slots correspond to excerpts of a sentence spoken by one single character, the first utterance usually ends with an hyphen, a comma or colon, and the second starts in lowercase.

The snippet of Figure 2 illustrates the aforementioned situation, and a rule has been designed to address it, resulting in the interaction:

```
(T3: And makes an offer so
     ridiculous, the farmer is
     forced to say yes.,
 A3: We gonna offer to buy
     Candyland?)
```

We refer to (Ameixa and Coheur, 2014) for additional details on other rules.

Finally, we note that the context of each turn is kept while building of the *Subtle* corpus. Although such context information is currently not used in the dialogue system described ahead, it is still kept as it may provide useful information for future improvements of the dialogue system. An excerpt of the resulting *Subtle* corpus is provided in Fig. 3.

```
SubId - 100000
DialogId - 1
Diff - 3715
T - What a son!
A - How about my mother?

SubId - 100000
DialogId - 2
Diff - 80
T - How about my mother?
A - Tell me, did my mother
      fight you?

SubId - 100000
DialogId - 3
Diff - 1678
T - Tell me, did my mother
    fight you?
A - Did she fight me?
```

Figure 3: Excerpt of the *Subtle* corpus obtained from the subtitle files.

In the example depicted in Fig. 3, `SubId` is a number that uniquely identifies the subtitle file from which the corresponding interaction was extracted. `DialogId` is a value used to find back-references to other interactions in the same conversation (the context). `Diff` is the difference in time (in milliseconds) between the trigger and the answer as registered in the subtitle file. Finally, T and A are the trigger and the answer, respectively. Note that, in the second interaction featured in the example of Fig. 3, it is very likely that both the trigger and the answer are spoken by the same character. This observation is also supported by the fact that the time difference between trigger and answer is very small. As already mentioned, the time difference will be taken into consideration when selecting the answer to an input by the user, both by weighting down answers with a time

difference that is too small (as in the example) or too large.

3.3 The Subtle Corpus: Some numbers

Table 1 summarizes some information regarding the *Subtle* corpus, generated when the time threshold between two slots is set to 0.

Table 1: Summarized information regarding the *Subtle* Corpus.

English

# Movies	# Movies ok	# Interactions	Average
5,764	5,665	5,693,811	1,005

Portuguese

# Movies	# Movies ok	# Interactions	Average
3,701	3,598	3,322,683	923

Some subtitle files did not comply with the usual `srt` format and were discarded. In English, from the initial 5,764 subtitle files (listed under **# Movies** in Table 1), 99 were discarded and only 5,665 files were used (listed under **# Movies ok** in Table 1). In Portuguese, from the initial 3,701 files, 3,598 were used to build the corpus. The processing of these files resulted in a total of 5,693,811 English interaction pairs (listed under **# Interactions** in Table 1) and 3,322,683 Portuguese interaction pairs, with an average number of interactions per file of 1,005 for English and 923 for Portuguese (**# Average** in Table 1).

4 The *Say Something Smart* Engine

In this section we describe the process of choosing an answer, being given an input from the user. When a user poses his/her request, this input is matched against the interactions in the *Subtle* corpus, and a set of answer candidates is retrieved. Then, a response needs to be chosen from the candidate answers. To this end, we index the *Subtle* corpus and extract a set of candidates; we score these candidates considering several measures and finally return the answer corresponding to the one attaining the best score.

In the continuation, we describe the indexing and selection process in further detail.

4.1 Corpora indexing and candidate extraction

Say something smart (SSS) uses Lucene[5] to index the *Subtle* corpus and its retrieval engine to obtain the first set of possible answers, given a user input (Ameixa et al., 2014). Lucene works with tokenizers, stemmers, and stop-word filters. We used the default ones for English, and the snowball analyzer for the Portuguese language.[6]

In the following we illustrate some of the retrieved interactions, considering the user input "*Do you have a brother?*":

```
(T4: You don't have to go,
       brother.,
 A4: I'm not your brother.)

(T5: You have a brother?,
 A5: Yeah, I've got a brother,
      man. You know that.)

(T6: Joe doesn't have a brother?,
 A6: No brother.)

(T7: Brother, do you have tooth
     paste?,
 A7: What brother?)

(T8: Have you seen my brother?,
 A8: He's not your brother
    anymore.)
```

The example above illustrates one of the problems of choosing an appropriate answer. As it can be seen, many of the interactions returned by Lucene have triggers that are not really related with the given input.

4.2 Choosing the answer

Given a user request u, Lucene retrieves from the set I of all interactions a subset U of N interactions, $U = \{(T_i, A_i), i = 1, \ldots, N\}$. Each interaction $(T_i, A_i) \in U$ is scored according to each of a total of four measures. The final score of each answer A_i to the user input u, $score(A_i, u)$, is computed as a weighted combination of the 4 scores $M_j, j = 1, \ldots, 4$:

$$score(A_i, u) = \sum_{j=1}^{4} w_j M_j(U, T_i, A_i, u), \quad (1)$$

where w_j is the weight assigned to measure M_j.[7]

The four measures implemented are described in the following.

[5] `http://lucene.apache.org`

[6] `http://snowball.tartarus.org/`

[7] All the measures to be applied and the associated weights can be defined by the user.

Trigger similarity with input The first measure, M_1, accounts for the *Jaccard similarity* (Jaccard, 1912) between the user input and the trigger of the interaction. For instance, given the input "*What's your mother's name?*", and the interactions:

```
(T9: How nice. What's your
     mother's name?,
 A9: Vickie.)

(T10: What was your
     mother's name?,
 A10: The mother's name
   isn't important.)
```

M_1 will assign a larger value to the second interaction, since "*What's your mother's name?*" is more similar to T10 than to T9, according with the Jaccard measure.

The measure M_1 is particularly important since, as previously discussed, many of the interactions returned by Lucene have triggers that have little in common with the given input. For example, and considering once again the previous input ("*What's your mother's name?*") some of the triggers retrieved by Lucene were:

```
T11: What's your name?

T12: What's the name your mother
   and father gave you?

T13: Your mother? how dare
   you to call my mother's name?.
```

Response frequency The second measure, M_2, targets the response frequency, giving a *higher score to the most frequent answer*. That is to say, we take into consideration the corpus redundancy. We do not force an exact match and the Jaccard measure is once again used to calculate the similarity between each pair of possible answers. Consider, for example, the request "*How are you?*" and the interactions:

```
T14: Where do you live?
A14: Right here.

T15: Where are you living?
A15: Right here.

T16: Where do you live?
A16: New York City.

T17: Where do you live?
A17: Dune Road.
```

M_2 will give more weight to the answer *Right here*, as it is more frequent than the others.

Answer similarity with input We also take into consideration the answer similarity (Jaccard) to the user input. Thus, M_3 computes the similarity between the user input and each of the candidate answers (after stop words removal). If scores are higher than a threshold it is considered that the answer shares too much words with the user input, and a score 0 is given to the answer; otherwise, the attained similarity result is used in the score calculus, after some normalisations.

Time difference between trigger and answer Finally, we can use in this process the time difference between the trigger and the answer (measure M_4). If there is too much time elapsed between the trigger and the answer, it is possible that they are not a real interaction.

$\diamondsuit$

To conclude, we refer that in (Ameixa et al., 2014) a hard-threshold is used to filter the interactions returned by Lucene considering a similarity measure; the most similar answer is used to decide which response is returned, much like our measure M_2. In this paper, we do not apply any hard-threshold. Instead, we combine a set of four different measures to score the candidates and select the one attaining the largets combined score.

5 Evaluation

In this section we describe some preliminary experiments conducted to validate the proposed approach.

5.1 Evaluation setup

Filipe, depicted in Fig. 4, is a chatbot previously built to allow user interactions with the SSS engine (Ameixa et al., 2014). It is on-line since November 2013.[8]

Using Filipe, we have collected a total of 103 requests made to the original SSS engine by several anonymous users. From this set, we removed

[8]It can be tested in `http://www.l2f.inesc-id.pt/~pfialho/sss/`

Figure 4: Filipe, a chatbot based on SSS.

the duplicates and randomly selected 20 inputs as a test set for our system.

5.2 Are subtitles adequate?

We started our evaluation with a preliminary inspection of *Subtle*, in order to understand if adequate responses could be found there. Three human annotators evaluated the first 25 answers returned by Lucene to each one of the 20 requests from the test set. For each request the annotators would indicate whether *at least one appropriate answer* could be found in these 25 candidate answers returned by Lucene.

The first annotator considered that 19 of the user requests could be successfully answered and that one could not, corresponding to the input "*What country do you live?*".

The second annotator agreed with the first annotator in 19 of the test cases. The only different test case corresponded to the input "*Are you a loser?*", to which the second annotator determined no suitable answer could be found in the ones returned by Lucene.

The third annotator disagreed with both annotators one and two with respect to the input "*What country do you live?*", as he considered "*It depends.*" to be a plausible answer. Additionally, this annotator considered that there was no plausible answer to the input "*Where is the capital of japan?*", to which the other two annotators agreed that "*58% don't know.*" was a plausible answer. Finally, the first and third annotators agreed that "*So what? You want to hit me?*", "*Your thoughtless words have made an incredible mess!*" or "*Shut up.*" would be appropriate answers to "*Are you a loser?*".

Despite the lack of consensus in these test cases, the fact is that the three annotators agreed that 17 out of 20 turns had a plausible answer in the set of answers retrieved by Lucene from the *Subtle* corpus, which is an encouraging result.

The next step is then to study the best way to select a plausible answer from the set of candidate answers retrieved by Lucene. Our framework, presented in Section 4, is evaluated in the continuation.

5.3 Answer selection

We tested five different settings $(S_1, \ldots, S_5)$ to score each interaction pair:

- S_1 – Only takes into account M_1.
- S_2 – Only takes into account M_2.
- S_3 – Takes into account M_1 and M_2.
- S_4 – Takes into account M_1, M_2 and M_3.
- S_5 – Takes into account all four measures.

For the settings S_{1-4} all measures considered were given the same weight. For S_5, however, the weights were optimized experimentally, yielding:

- 40% weight for M_1.
- 30% weight for M_2.
- 20% weight for M_3.
- 10% weight for M_4.

The test set described in Section 5.1 was again used, and SSS was tested in each of the five settings $S_1, \ldots, S_5$ described above. The best scored answer of each log was returned.

In order to evaluate how plausible the returned answers were, a questionnaire was built. It contained the 20 user request from the test set and the answers given considering each of the settings (duplicate answers were removed). We told the evaluators that those were the requests posed by humans to a virtual agent and the possible answers. They should decide, for each answer, if it made sense or not. Figure 5 shows an extract of the questionnaire. 21 persons filled the questionnaire. Results are summarized in Table 2.

	S_1	S_2	S_3	S_4	S_5
%	39.29	45.24	46.90	61.67	51.19

Table 2: Percentage of plausible answers in each setting.

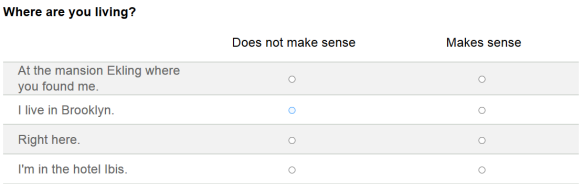

Figure 5: Example of a question in the questionnaire.

We can see that the S_2 setting achieved better results than S_1, and that S_3 (the combination of measures M_1 and M_2) achieved slightly better results than the previous two. This suggests that the combination of the two strategies may yield better results than any of them alone. Moreover S_4 (which added the third measure M_3) achieved the best results, with a difference of almost 15% compared to the strategy of S_3. The last setting (which added the M_4 measure), however, achieved worse results than S_3.

To conclude, our preliminary evaluation suggests that the similarity between the user request and the trigger and the answer should be considered in this process, as well as the redundancy of the answers.

6 Conclusions and future work

As it is impossible to handcraft responses to all the possible OOD turns that can be posed by humans to virtual conversational agents, we hypothesise that conversations between humans can provide some plausible answers to these turns.

In this paper we focus on movies subtitles and we describe the process of building an improved version of the *Subtle* corpus, composed of pairs of interactions from movies subtitles. A preliminary evaluation shows that that the *Subtle* corpus does include plausible answers. The main challenge is to retrieve them. Thus, we have tested several measures in SSS, a platform that, given a user input, returns a response to it. These measures take into consideration the similarities between the user input and the trigger/answer of each retrieved interaction, as well as the frequency of each answer. Also, the time elapsed between the subtitles is taken into consideration. Different weights were given to the different measures and the best results were attained with a combination of the measures: 21 users considered that 61.67% of the answers returned by SSS were plausible; the time elapsed between the turns did not help in the process.

There is still much room from improvement. First, the context of the conversation should be taken into consideration. An automatic way of combining the different measures should also be considered, for instance using a reinforcement learning approach or even a statistical classifier to automatically estimate the weights to be given to each measure. Moreover, semantic information, such as the one presented in synonyms, could be used in the similarity measure; information regarding dialogue acts could also be used in this process.

Also, responses that refer to idiosyncratic aspects of the movie should receive a lower score. Although M_2 can be seen as an indirect metric for this domain-independence (a frequent response is less likely to come with a strong contextual background), responses that include names of particular persons, places or objects should be identified. However, this strategy is not straightforward, as some particular responses containing named entities should not be discarded. This is the case not only to address factoid questions, but also for inputs such as "*Where do you live?*" or "*What is your mother's name?*" whenever a pre-defined answer was not prepared in advance.

Currently we are collecting characters' language models, and intend to use these during the answer candidate selection. Additionally, we are in the process of combining information from movie scripts to enrich subtitles, for example by adding in character names. This added information would enable an easier identification of the dialogue lines of each character as well as creating specific language models; finally, this could also allow us to filter some interaction pairs that represent two lines from the same character.

Acknowledgements

We would like to acknowledge the administrator of OpenSubtitles for providing the subtitle data used in this paper and the anonymous reviewers for helpful comments. This work was partially supported by EU-IST FP7 project SpeDial under contract 611396 and by national funds through FCT – Fundação para a Ciência e a Tecnologia, under projects PEst-OE/EEI/LA0021/2013 and CMUP-ERI/HCI/0051/2013.

References

David Ameixa and Luísa Coheur. 2014. From subtitles to human interactions: introducing the subtle corpus. Technical report, INESC-ID.

David Ameixa, Luisa Coheur, Pedro Fialho, and Paulo Quaresma. 2014. Luke, i am your father: dealing with out-of-domain requests by using movies subtitles. In *Proceedings of the 14th International Conference on Intelligent Virtual Agents (IVA'14)*, LNCS/LNAI, Berlin, Heidelberg. Springer-Verlag.

Rafael Banchs and Haizhou Li. 2012. IRIS: a chat-oriented dialogue system based on the vector space model. In *Proc. 50th Annual Meeting of the ACL: System Demonstrations 50th Meeting ACL (System Demonstrations)*, pages 37–42.

Rafael E. Banchs. 2012. Movie-DiC: a movie dialogue corpus for research and development. In *Proc. 50th Annual Meeting of the ACL: System Demonstrations 50th Meeting ACL (Short Papers)*, pages 203–207.

Niels Ole Bernsen and Laila Dybkjaer. 2005. Meet Hans Christian Anderson. In *Proc. 6th SIGdial Workshop on Discourse and Dialogue*, pages 237–241.

Timothy Bickmore and Justine Cassell. 2000. How about this weather? social dialogue with embodied conversational agents. In *Socially Intelligent Agents: The Human in the Loop*, pages 4–8.

Pedro Fialho, Luísa Coheur, Sérgio Curto, Pedro Cláudio, Ângela Costa, Alberto Abad, Hugo Meinedo, and Isabel Trancoso. 2013. Meet Edgar, a tutoring agent at Monserrate. In *Proc. 51st Annual Meeting of the ACL: System Demonstrations*, pages 61–66, August.

Matthew Henderson, Colin Matheson, and Jon Oberlander. 2012. Recovering from Non-Understanding Errors in a Conversational Dialogue System. In *Workshop on the Semantics and Pragmatics of Dialogue*.

P. Jaccard. 1912. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50.

Ronakkumar Patel, Anton Leuski, and David Traum. 2006. Dealing with out of domain questions in virtual characters. In *Proceedings of the 6th International Conference on Intelligent Virtual Agents*, IVA'06, pages 121–131, Berlin, Heidelberg. Springer-Verlag.

Thies Pfeiffer, Christian Liguda, Ipke Wachsmuth, and Stefan Stein. 2011. Living with a virtual agent: Seven years with an embodied conversational agent at the Heinz Nixdorf MuseumsForum. In *Proc. Re-Thinking Technology in Museums 2011*, pages 121–131.

David Traum, Priti Aggarwal, Ron Artstein, Susan Foutz, Jillian Gerten, Athanasios Katsamanis, Anton Leuski, Dan Noren, and William Swartout. 2012. Ada and Grace: Direct interaction with museum visitors. In *Proc. 12th Int. Conf. Intelligent Virtual Agents*, pages 245–251.

Ulli Waltinger, Alexa Breuing, and Ipke Wachsmuth. 2011. Interfacing virtual agents with collaborative knowledge: Open domain question answering using Wikipedia-based topic models. In *Proc. 22nd Int. Joint Conf. Artificial Intelligence*, pages 1896–1902.