A Comparison of Services for Intent and Entity Recognition for Conversational Recommender Systems

Andrea Iovine
University of Bari Aldo Moro
Bari, Italy
andrea.iovine@uniba.it

Fedelucio Narducci
Politecnico di Bari
Bari, Italy
fedelucio.narducci@poliba.it

Marco de Gemmis
University of Bari Aldo Moro
Bari, Italy
marco.degemmis@uniba.it

Marco Polignano
University of Bari Aldo Moro
Bari, Italy
marco.polignano@uniba.it

Pierpaolo Basile
University of Bari Aldo Moro
Bari, Italy
pierpaolo.basile@uniba.it

Giovanni Semeraro
University of Bari Aldo Moro
Bari, Italy
giovanni.semeraro@uniba.it

ABSTRACT

Conversational Recommender Systems (CoRSs) are becoming increasingly popular. However, designing and developing a CoRS is a challenging task since it requires multi-disciplinary skills. Even though several third-party services are available for supporting the creation of a CoRS, a comparative study of these platforms for the specific recommendation task is not available yet. In this work, we focus our attention on two crucial steps of the Conversational Recommendation (CoR) process, namely Intent and Entity Recognition. We compared four of the most popular services, both commercial and open source. Furthermore, we proposed two custom-made solutions for Entity Recognition, whose aim is to overcome the limitations of the other services. Results are very interesting and give a clear picture of the strengths and weaknesses of each solution.

Author Keywords

conversational recommender systems; natural language processing; conversational agents

CCS Concepts

•**Computing methodologies** → **Discourse, dialogue and pragmatics; Information extraction;** •**Information systems** → **Recommender systems;** •**Human-centered computing** → **Natural language interfaces;** Please use the 2012 Classifiers and see this link to embed them in the text: https://dl.acm.org/ccs/ccs_flat.cfm

INTRODUCTION

Conversational Recommender Systems (CoRSs) are intelligent systems that provide personalized access to large sets of items (e.g. e-commerce catalogues, daily news, streaming platforms, etc.) by exposing a *conversational interface*. Indeed, the distinguishing feature of a CoRS compared to a standard recommender system is its ability to interact with the user during the whole recommendation process. Chat-based interfaces have been often proposed as a way to bring interactivity into the recommendation process: the user and the recommender system interact by exchanging messages in natural language. However, developing a chat-based interface for a CoRS is a very challenging task, since it requires multi-disciplinary skills: Natural Language Processing (NLP), Machine Learning (ML), and Human-Computer Interaction (HCI). The scope of this work is to analyze the main NLP steps performed by a CoRS. In this area, two main Natural Language Understanding (NLU) tasks play a crucial role: Entity Recognition (ER) and Intent Recognition (IR). ER consists of identifying real-world entities that users can refer to during the dialogue. A CoRS needs to acquire preferences from users to build their profile, and being able to recognize items and properties is essential in order to achieve high-quality recommendations. The IR step is equally important: if the user's request is not identified correctly, the entire recommendation process may fail, resulting in user frustration, and low user retention rates. Many third-party NLU services are available, which can perform both ER and IR tasks. However, an extensive evaluation of these solutions in the area of CoRSs has not been done yet. A relevant work on that direction has been proposed by [4]. The authors evaluated different platforms on a corpus of questions related to public transportation, and on another one from two StackExchange platforms. We decided to investigate ER and IR tasks in the music, book, and movie recommendation scenarios. In our opinion, those scenarios are really challenging for several reasons: entity names often are complete sentences with their own meaning (e.g. the movie title "Life is beautiful"); aliases are often associated to person names (e.g. the American song-writer *Barry Eugene Carter* also known as *Barry White*); intents can be expressed in many different forms (e.g. for recommendation requests, users can send messages like *Suggest me a movie*, *What can I watch tonight?*, *I'm looking for a sci-fi movie*). In this study, we

IntRS '20 - Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, September 26, 2020, Virtual Event

evaluated four third-party services. For the ER task, we also added two of our own solutions, for a total of six systems. The experiment has been carried out on a dataset of real user messages, collected from the interactions with a CoRS in the book, music, and movie domains. Therefore, we evaluate each platform's ability to understand messages written by real users in the recommendation context. The experiment was carried out in order to answer the following Research Questions:

RQ1 - Is a purpose-built solution better than a general-purpose one for a chat-based CoRS? Developing a customized component requires more effort, however this effort may not result in better performance.

RQ2 - Does fuzzy matching improve Entity Recognition performance? Introducing Fuzzy matching for the ER step might be a solution to the problem of identifying misspelled or incomplete entity mentions. However, it may also generate some noise in the recognition process (e.g. treating unrelated words as entities).

RQ3 - Is a commercial Intent Recognizer better than an open-source one for a chat-based CoRS? Taking inspiration from [4], commercial platforms can exploit large quantities of data fed by their userbase. Conversely, open-source solutions like Rasa can only rely on the data that the current user is able to provide for training. This hypothetically means that the former has better performance than the latter. At the end of this work, we highlight the specific peculiarities of each service, and we explain why one service might be preferred over another via a detailed analysis of specific cases.

RELATED WORK

A CoRS is defined as a system that provides recommendations to users via a multi-turn dialogue [14]. The main characteristic of CoRSs is that the user profile is acquired in an iterative fashion. The system can interact by asking the user to rate some items, and the user can influence the outcome of the recommendation by providing feedback on the suggested items. On the other hand, traditional recommender systems require that all user information is provided before generating a recommendation [18]. A review of state of the art regarding Conversational Recommender Systems is described in Jannach et al. [14]. Kang et al. [15] investigated how people interact with a natural language-based CoRS through voice or text. To do this, the authors developed a natural language interface, and integrated it in the MovieLens system. The authors classified three types of recommendation goals and several types of follow-up queries from the collected data. Objective recommendation goals express the desire to find movies based on known attributes, such as the genre. Subjective goals instead involve the evaluation of subjective qualities (e.g. "sad movies", "interesting characters"). Finally, navigational goals are expressed when the user wants to find a specific movie by its title. A work similar to this is described in Cai and Chen [5]. An exploratory study was conducted by manually annotating 200 human-human conversations. The result is a taxonomy of intents that are commonly associated with the task of receiving recommendations. The authors state that users can express preferences by either mentioning entities (e.g. movies that they have previously watched), attributes (i.e. subjective or objective qualities of the items), or purposes (e.g. "a movie to watch with the family"). While these works are focused on understanding how people express their needs to a CoRS, the scope of our study is to evaluate the ability of the existing NLU tools to correctly understand these expressions.

Developing a CoRS that interacts using a natural language dialogue requires performing several Natural Language Processing tasks. Most NLP strategies must be trained on large quantities of text. In recent years, researchers have published many datasets for training and evaluating CoRSs. Examples are Dodge et al. [9], Asri et al. [1], Suglia et al. [24], Li et al. [16], and Radlinski et al. [21]. However, many of the datasets in the literature were developed with the specific intent of training End-to-End systems, i.e. systems that learn the two tasks of maintaining a conversation and generating recommendations together. In this paper, we only focus on the first task so that the underlying recommendation can be performed with any of the existing algorithms. As stated in the Introduction, several NLU platforms are already available, which contain many of the essential blocks needed to quickly design and develop Conversational Agents (CAs). They extract relevant information for a specific machine understanding task from natural language content. [8]. These systems have been demonstrated to be suitable for several NLP tasks [25] such as question answering [22], document translation [23], and dialogue management [2]. Entity Recognition and Intent Recognition are tasks that are frequently addressed by those systems. Google's Dialogflow (also known as API.ai)[1], IBM's Watson Assistant[2], Microsoft's LUIS[3] are the most famous commercial solutions. Other platforms are Amazon Lex[4], Facebook's Wit.ai[5], and Recast.ai[6]. On the other hand, open-source services like Snips.ai[7] and RASA[8] are also available. These systems have been largely described and analyzed in the literature, showing encouraging performance on different datasets [27, 7, 4, 6, 17]. As for NLU systems for conversational agents, Wisniewski et al. [27] compares several systems with the aim of identifying the best solution for building a robust CA. The authors compared Snips.ai with Amazon Lex, API.ai, LUIS, Apple's SiriKit[9], IBM Watson, Wit.ai on a dataset of 328 queries chosen by the Snips.ai team. The results confirm the validity of the service, showing much higher accuracy values than its competitors for the Intent Recognition task. Braun et al. [4] compared the performance obtained by LUIS, Watson Assistant, API.ai, Wit.ai, and Amazon Lex on two different datasets. The first one consists of 206 questions about public transport, which were asked to a Telegram chatbot, and then manually annotated by the authors. The second one consists of 290 questions made on two StackExchange

[1] https://dialogflow.cloud.google.com/
[2] https://www.ibm.com/cloud/watson-assistant/
[3] https://www.luis.ai/
[4] https://aws.amazon.com/it/lex/
[5] https://wit.ai/
[6] https://cai.tools.sap/
[7] https://snips.ai/
[8] https://rasa.com/
[9] https://developer.apple.com/documentation/sirikit

platforms. The results show that LUIS performs better than its competitors on the two datasets in terms of F1-score.

The state of the art shows that the research area of CoRSs is indeed quite active, and that many works have been published on the subject. However, very few of them focus on the evaluation of Natural Language Understanding components for the conversational recommendation scenario. We believe that this is important, especially if the CoRS should support a *mixed-initiative* interaction, in which both the system and the user can take control of the conversation. Furthermore, we believe that this research is interesting for practitioners that would like to develop a conversational recommender system using tools that are currently available on the market. Research shows that they perform well in domains such as question answering and customer service, but there is no evidence regarding their performance in the conversational recommendation scenario. We believe that there are some challenges that are unique to this application. In fact, the intents sand entities of the domains explored in [4] are not particularly ambiguous (e.g. Station destination, Station start, Departure time, etc.). Conversely, the conversational recommendation scenario offers both ambiguous entities, and user requests that are highly heterogeneous, as anticipated in the Introduction.

CORS ARCHITECTURE AND DIALOG MODEL

Chat-based CoRSs are a specialization of a Goal-Oriented Conversational Agent. A general architecture for CAs can be found in Williams et al. [26]. A possible adaptation of the Goal-Oriented CA architecture for the chat-based CoRS task can be seen in Fig. 1. For this paper, we exclude the components related to speech, since the focus is on written text messages. The user message is first analyzed by the *Natural Language Understanding* (NLU) component. This component is responsible for extracting all information from the user message. Two tasks are commonly associated with Natural Language Understanding: *Intent Recognition* and *Entity Recognition*. Intent Recognition is performed to understand the general action or request that is formulated in the message. For example, asking *"What movie should I watch tonight"* is a clear indication of the user's intention of receiving recommendations from the system. Entity Recognition is used to extract named entities that are mentioned by the user. This is essential for a CoRS, as it allows the user to discuss about the items that he or she is interested in. A CoRS that suggests movies should be able to recognize entities such as movies, actors, directors, genres, etc. The output of the NLU component is a semantically tagged version of the user message, which describes the intent of the user, and the ratings to each individual entity that may have been mentioned within. The next module in the architecture is the *Dialogue State Tracker* (DST). It is responsible for keeping a consistent conversation between the user and the system. It also remembers all the information that was exchanged between the two parties, which is referred to as the *dialogue state*. The DST makes it possible for the system to remember what was said previously in the conversation, e.g. when the user answers a question made by the system. The state is updated at each conversation turn, based on the user's message and the previous state. Several DST strategies have been proposed in the literature: rule-based, such as Branting et al. [3]; frame-based, such as Göker and Thompson [12]; based on statistical models, such as Gavšić and Young [11]; and based on neural networks, such as Bordes et al. [2]. For the CoRS task, the dialogue state can contain the user's profile. Frame-based DST is often used for this purpose, as it models the conversation as a set of *slots* (i.e. attributes) that need to be *filled* by the user with the necessary information. An example of this is the Adaptive Place Advisor [12], which uses natural language to model a constraint-based recommender system. The *Dialogue Policy* (DP) is in charge of choosing the action that will be performed by the system, based on the current input and the state of the conversation. Policy selection can be performed either via rules (e.g. one action per intent), or can be selected via a probabilistic model, such as [11]. The actions that a CoRS can perform are mostly related to the profile acquisition (e.g. ask the user to rate an item), or recommendation (e.g. generate a list of recommendations). Therefore, the DP can directly communicate with the underlying recommender system. It is worth noting that the architecture described in Fig. 1 is not bound to a specific recommendation algorithm. Finally, the *Natural Language Generator* (NLG) is in charge of generating the textual response that will be sent back to the user. This response can be a feedback of an action made by the system, or a question that the user should answer, or a generic response to a colloquial message. The simplest method for NLG is to use *response templates* that can be filled with contextual data. End-to-End conversational agents such as [2] attempt to generate a complete sentence based on the user's input. In this paper, we focus our attention on the NLU component, and in particular, on the Intent and Entity Recognition tasks, which we introduced in Section 1.

To model the interaction between the user and the CoRS, we devised a general *dialogue model*, which involves three distinct actions: providing a preference to one or more items, requesting a recommendation, requesting to view her profile. We chose these three actions as they represent the most basic activities that are performed by recommender systems: acquiring the user profile and generating recommendations. We then added the ability to show the user profile, which is essential for ensuring transparency. Another advantage of this dialog model is that it is agnostic to the underlying recommendation algorithm, as it can be easily adapted to work with a collaborative or content-based filtering recommender system. Finally, it is consistent with the findings of [15] and [5]. In the movie domain, given the utterance *I like Ghostbusters because I love comedy movies*, the CoRS should recognize that the user is providing a positive preference for two different kinds of items: a movie (*Ghostbusters*) and a genre (*comedy*). By asking the system *Can you suggest me a movie to watch?* instead, the CoRS should be able to understand that the user is asking for a recommendation. Finally, the message *What are the preferences stored in my profile?* means that the user is asking for exploring his/her profile.

EVALUATED PLATFORMS

The objective of this study is to compare several NLU platforms based on their ability to perform tasks commonly carried out in a chat-based CoRS, i.e. intent and entity recognition. Due to the large quantity of NLU platforms available on the

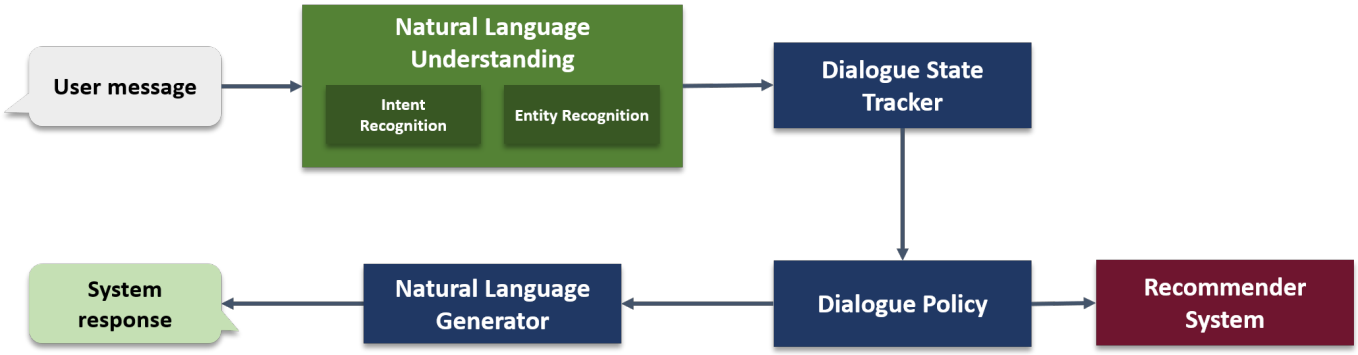

Figure 1: Possible architecture for a chat-based CoRS

market and limited resources, a comparison of all platforms would hardly be feasible. Therefore, we decided to conduct the study on a subset of the most popular platforms. We chose to include three of the most popular commercial NLU platforms, which are Google Dialogflow, Microsoft LUIS, and IBM Watson Assistant. Then, we selected one of the most popular open-source NLU platforms, i.e. Rasa. These four platforms already provide all the tools required to build a functioning CoRS, except for the recommendation algorithm. On the surface, the commercial platforms share many similarities: they all provide the same NLU functionalities (i.e. intent and entity recognition), a Web-based interface for building the dialogue model, multi-language support, and can be easily integrated into a wide variety of messaging platforms. Also, they all employ Machine Learning (ML) algorithms for the NLU tasks, which need to be trained by supplying example sentences. The ML techniques used are proprietary, and thus no implementation details are provided, which makes it more difficult to choose one platform over another based on their characteristics. An advantage of Dialogflow and Watson Assistant over LUIS is the *fuzzy matching* function that allows the Entity Recognition to tolerate spelling errors. Rasa differs from the previous platforms for several reasons: first and foremost, it is open-source and self-hosted, rather than proprietary and cloud-based. Rasa is essentially divided into two parts: Rasa NLU, which handles the Natural Language Understanding pipeline, and Rasa Core which implements a Dialogue management component. This component uses ML to model the conversation logic, unlike the previous platforms that use hand-written rules. Due to the open-source nature of Rasa, it is also more transparent about the ML techniques that it employs. In fact, each component of the NLU pipeline can be customized, and many state-of-the-art models are already available, such as spaCy embeddings. However, Rasa does not currently implement a fuzzy matching function for Entity Recognition. While a developer could plug an external fuzzy string matching library for this scope, we chose not to do it, as we want to test Rasa's performance *out of the box*. In addition to these platforms, we also added to the comparison of two Entity Recognizers that were developed in-house, and that is purpose-built for the CoRS task. Both components are trained to recognize and link entities that are contained within a knowledge base.

This composition is tailored to the Research Questions formulated in the Introduction. We included both commercial and open-source platforms in the study to test the hypothesis formulated in RQ3. The addition of custom-built components for Entity Recognition is useful to answer RQ1. Finally, the fact that only some platforms support Fuzzy Matching for Entity Recognition allows us to answer RQ2.

Custom ER solutions

Before starting the description of the custom-made ER models, it is worth noting that the entity recognition task for a CoRS is not trivial. Indeed, despite the fact that entity names are already available in the knowledge base, the ER should be able to tolerate misspellings, incomplete mentions, and synonyms. In fact, when a user interacts with a conversational agent thorough text, typing errors are likely to occur [19]. Another problem is that the same entity may have several surface forms. One example of such is Barry White as introduced in Section 1. Furthermore, users may expect to mention entities in a shortened form, for convenience. An efficient ER should properly manage these issues, and not accounting for them will result in a more frustrating user experience.

NER_FCR (Named Entity Recognition based on Fuzzy CRF + Regex) is composed of two parts: an *Entity Recognizer* that extracts the parts of the user message containing entity mentions, and an *Entity Linker* that maps the entity mentions back to the knowledge base. The Entity Recognition is implemented using parts from the Stanford CoreNLP[10] library. In particular, it uses a combination of two algorithms: one based on *regular expressions* and one based on *Conditional Random Fields* (CRF) [10]. The CRF-based classifier can be trained by supplying example sentences. It classifies each word as either entity or non-entity, based on several features such as the shape and Part-of-speech tag of the current, previous, and next word. It is able to tag entities even if they were not spelled correctly or completely. The regex-based classifier only needs a list of entities, and ensures that correctly spelled entities are recognized. The Entity Linking part uses fuzzy string matching[11] to find the entities that are most similar to the mention based

[10] https://stanfordnlp.github.io/CoreNLP/
[11] https://github.com/xdrop/fuzzywuzzy

on the Levenshtein distance. Multiple entities are retrieved if the mention is ambiguous.

NER_KG (Named Entity Recognition based on Knowledge Graphs) exploits a knowledge-based approach for finding entities mentioned in the user sentence and linking them to the correct concept in the knowledge base. Wikidata[12] is the Knowledge Base chosen for this task. In particular, we can customize NER_KG according to the entities involved in a specific domain. For example, if the CoRS is a movie recommender system, we can filter only Wikidata concepts related to this specific domain: e.g. movies, actors, and directors. NER_KG performs both *Spotting* and *Linking* steps. In the spotting step, the algorithm analyzes the text in order to discover candidate entities. In particular, the algorithm detects sequences of words (*surface form*) matching a Wikidata alias, and then all the concepts that can be associated to the alias are retrieved. This spotting strategy enables NER_KG to deal with noisy text, For the second step (also referred to as *Linking*), it uses an approach based on graph embeddings to exploit the relations between concepts in the knowledge graph. In particular, NER_KG uses holographic embeddings (HolE) [20] built on Wikidata. The disambiguation step consists of selecting the correct concept for each surface form. The idea is to choose the concept that is more similar to the other concepts occurring in the text, following the hypothesis of *one topic for discourse*. The motivation behind this approach is that, in a sentence, the user tends to refer to entities that are in some way related. Due to the use of linked data, NER_KG does not require training sentences to work.

DATASET

The dataset used in this study contains real user messages, which were captured by logging the interactions between users and a chat-based CoRS in the movie, book, and music domains [13]. Each message is bundled with the ID of the user, the intent, the entities mentioned within, and a timestamp. The dataset is composed of 5,318 messages for the movie domain, 1,862 for the book domain, and 2,096 for the music domain. The dataset was filtered by removing all messages that do not fit with the dialogue model defined in Section 3. For instance, we removed generic chit-chat messages, or messages that users sent in response to a question made by the system. The remaining messages have been manually checked by three human annotators. In particular, the annotators validated the intents identified by the CoRS, checked the list of entities identified in the message, and fixed errors. For each entity mentioned by the user, the annotators tagged it with the most appropriate Wikidata entity.

At the end of the cleaning and revising process, 2,410 queries have been obtained. In particular, there are 1,016 messages for the movie domain, 747 for the book domain, and 647 for the music domain. Table 1 presents an overview of the composition of the final dataset. Table 2 contains some examples of user messages in the movie domain. Each message is annotated with one of the following intents: *preference, request recommendation, and show profile*, which are the actions of the dialogue model described in Section 3. Each entity was manually classified according to its role in the domain. The roles defined for each domain are: *item_movie, item_song, item_book* for entities concerning movies/songs/books; *item_people* for entities concerning real people like actors, singers or writers; *item_genre* that are entities describing the genre of an item. Entities are all linked to their corresponding Wikidata ID. The dataset is publicly available on GitHub[13].

Dataset	Intent	∑		Entity	∑
movie	preference	743		item_movie	460
	request_recommendation	206		item_people	289
	show_profile	67		item_genre	27
music	preference	534		item_song	85
	request_recommendation	146		item_people	414
	show_profile	67		item_genre	37
book	preference	438		item_book	273
	request_recommendation	142		item_people	149
	show_profile	67		item_genre	18

Table 1: Details on the composition of the final datasets

EXPERIMENTAL EVALUATION

The objective of this experiment is to evaluate the performance of the NLU platforms presented in Section 4 for the Intent and Entity recognition tasks in the Conversational Recommendation (CoR) scenario. We used the datasets described in Section 5 both to train and test the ML components of each platform. The idea is that by performing the test on multiple domains we will obtain stronger evidence about the strengths and weaknesses of each solution. Moreover, we are interested in observing each platform's ability to understand messages sent by real users in the CoR scenario. This scenario is challenging due to the high variability of user messages, and to the presence of misspelled or partial entity mentions. Although a service may perform better in a domain and worse in others, we expect that all platforms show similar performance in all three domains.

Protocol

The experimental protocol adopted in our work is inspired from [4], with some changes. First, we adopted a 5-fold cross-validation instead of a random train-test split. Also, we decided to analyze IR and ER performance separately. All NLU platforms described in Section 4 followed the same experimental protocol, which means that each platform was trained and tested using the same data. Two exceptions are made: first, only the ER test was performed for NER_FCR and NER_KG. Also, NER_KG does not require training sentences, so cross-validation was unnecessary. To perform the 5-fold CV, we divided each dataset into five parts. We then repeated training and testing five times, by using one fold as a test set, and the other four folds as training set. In each step, we trained the Intent and Entity Recognition components of each platform with the training set. Where possible, we used the batch import functionalities made available by each platform, in order to easily re-train the agents for each iteration. During testing, the user messages in the test set were

[12] https://www.wikidata.org/

[13] https://github.com/aiovine/converse-dataset

Message	Intent	Entities
I like the avengers	preference	"The Avengers (Q182218)" - positive
I like ghostbuster	preference	"Ghostbusters (Q108745)" - positive
I don't like blade runner	preference	"Blade Runner (Q184843)" - negative
I like Titanic, but I don't really like James Cameron	preference	"Titanic (Q44578)" - positive "James Cameron (Q42574)" - negative
i like Jonny Depp	preference	"Johnny Depp (Q37175)" - positive
See my profile	show_profile	
Suggest some film	request_recommendation	
Can you suggest me an action movie?	request_recommendation	

Table 2: Examples of user messages contained in the movie dataset

analyzed by the platform. For LUIS, we used the integrated testing function. For the other platforms, we created a script that calls the available APIs programmatically. The intent and entities that were extracted by the system were compared with the actual tags in the dataset. For the IR task, an error was counted if the recognized intent was different from the true intent, or if no intent was found by the system. For the ER task, a message was considered correctly tagged if all entity mentions contained within were found, and if each mention was linked to the correct entity in the knowledge base. Therefore, we counted an error when an entity mention was not captured by the system, when one or more words were incorrectly tagged as entity mentions, or when the entity mention was successfully found, but it was linked to the wrong entity. Also, if an entity was recognized twice in the same sentence, one error was recorded. It is worth noting that each platform received the entire set of possible entities in the training phase, because measuring the ability to recognizing previously unseen entities is out of the scope of this experiment. Since Rasa offers multiple choices in terms of NLU pipelines, we also conducted a preliminary test in order to choose, for each domain, the pipeline that obtained the highest F1-score for both Intent and Entity Recognition. As a result, we used the *supervised_embeddings* pipeline for the movie domain, and the *pretrained_embeddings_spacy* pipeline for the book and movie domains.

In order to measure the performance, we used the *precision*, *recall* and *F1-score* metrics. These metrics were calculated for the Intent and Entity Recognition tasks separately. Aside from the per-domain measures, we also calculated the overall measures, aggregated from the three domains. We decided to exclude the precision of the *intent-not-found* class from the calculation, because we do not find it relevant to our study. Therefore, whenever a platform did not return any intent for a message, only a false negative has been recorded.

Results

The results of the experiment can be found in Figure 2 and Tables 6 - 11. Each table reports the results of the Intent and Entity Recognition tests for a single NLU platform. We reported the values of the three metrics for each individual intent/entity type. The Total row is calculated over all intent or entity types in a domain. The Overall row is calculated over all domains. Figure 2 describes the F1-score recorded respectively for the Intent and Entity Recognition test by each platform, in each domain, plus an overall figure that summarizes all three domains. Observing the results, all platforms behave relatively well in terms of Intent Recognition, with F-scores no lower than 0.97. A notable example is Rasa, which obtained a perfect IR score in the music domain. Overall, Watson Assistant is the platform that obtains the best performance, with an F1-score of 0.9988. However, by analyzing the trend in each domain, we see that the situation is not as clear: Watson, LUIS, and Rasa trade places as the best system in each domain. Instead, Dialogflow seems to perform the worst, with an overall F1-score of 0.9908.

For the Entity Recognition task, the best two systems are NER_FCR and Dialogflow, with an overall F-score of 0.9920 and 0.9911, respectively. In second place we find Watson Assistant, with an F-score of 0.9715. In the last three places, we find NER_KG, LUIS, and Rasa, that obtained similar results throughout the domains, with the lowest score achieved by Rasa on the book domain with 0.8399. Watson Assistant seems to struggle most with recognizing genres, with a recall of 0.6111 and an F-score of 0.7586 in the book domain, as we can see in Table 6. LUIS obtained mostly high precision and low recall for each entity type, as seen in Table 8. This is expected due to the lack of fuzzy matching, which means that only exact entity matches will be recognized. A similar trend can be seen for Rasa in Table 9, which makes sense because it also lacks fuzzy matching.

In particular, Rasa has some difficulties in recognizing movie names, music artists, and genres (especially in the book domain), with a recall of 0.3889 and an F-score of 0.56. Unlike the previous platforms, Dialogflow is able to handle all entity types without major problems. In fact, it was able to recognize all song entities correctly. This is also true for NER_FCR, which performed comparably to Dialogflow for every entity type. NER_KG was able to recognize genres better than the other platforms (with F-scores ranging from 0.8955 to 0.9811), however it fared worse than the others in recognizing movies, songs, and book names. It especially struggled with recognizing songs, obtaining a recall of 0.6786 and an F-score of 0.7917.

A one-way ANOVA statistical test (significance level = 0.05) was performed. The F-score was the dependent variable, with the platform being the independent variable. The null hypothe-

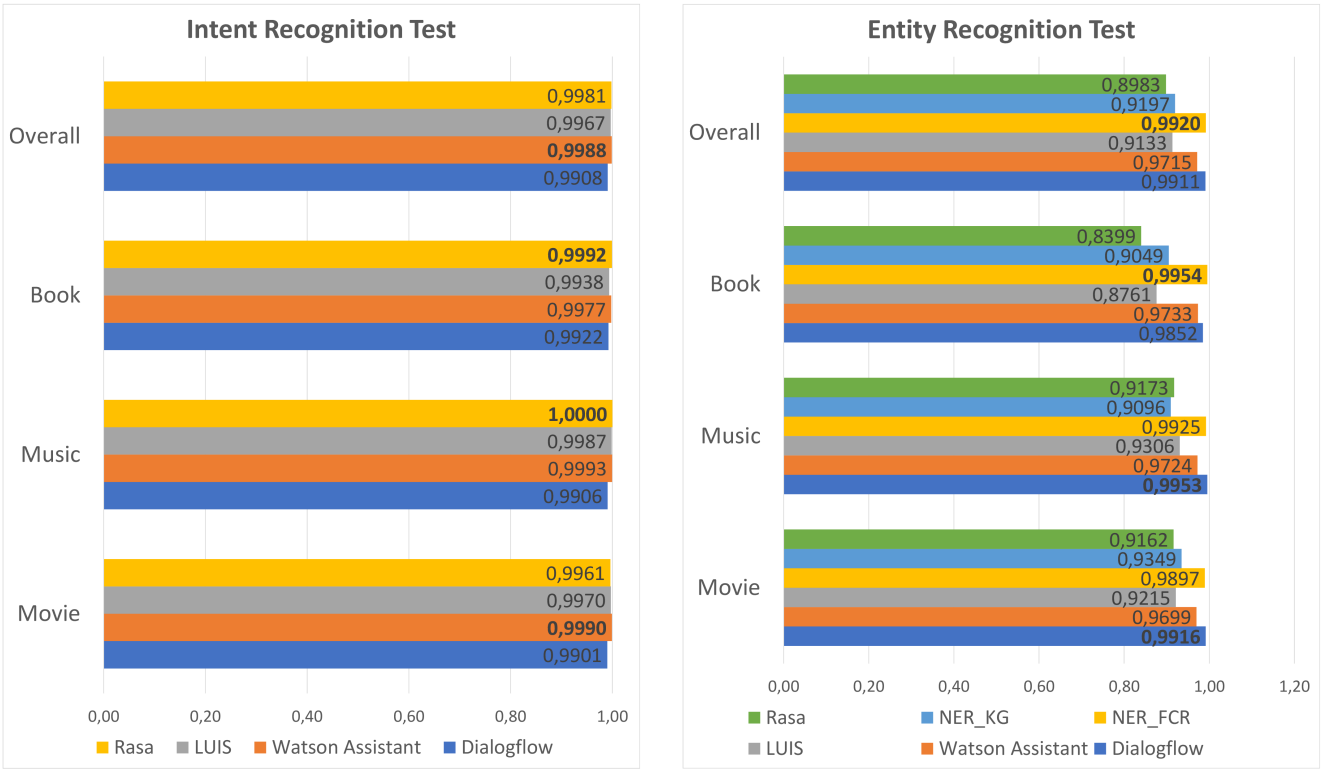

Figure 2: F1-score results for the Intent (left) and Entity (right) Recognition tests

sis for this test is: *The average F-score for Intent/Entity recognition is the same for all platforms*. Results of the ANOVA test for both intents and entities are described in Table 3. The test rejected the null hypothesis in both cases. Therefore, we proceeded to perform post-hoc tests, to find which platforms perform better (or worse) than others.

To do this, we used the *t-test for dependent samples* (significance level = 0.05). The results of the t-test for the Intent Recognition tasks are described in Table 4. For each pair of platforms, we reported the value of the *t* statistic and the p-value. If the p-value is less than the significance level, a symbol is added: (+) means that the platform on the row has a significantly higher F-score than the platform on the column, (-) means the opposite. The t-test did not find a clear winner for the IR task. However, it confirmed that Dialogflow was significantly worse than the other three platforms, which is in accordance with what was said at the beginning of this Section. More interesting results come from the ER task represented in Table 5. In fact, almost all pairs of platforms obtained significantly different results. Therefore, we can confirm that Dialogflow and NER_FCR both obtained the best performance. Watson Assistant placed second, followed by NER_KG, then LUIS, then Rasa.

Discussion

With the results described in the Section 6.2, we can answer the Research Questions introduced in the Introduction. Regarding RQ1, we can see that one of our own components (NER_FCR)

	Intent Recognition test			**Entity Recognition test**	
Domain	**F**	**p**		**F**	**p**
Total	18.61	<0.001		129.8	<0.001
Movie	6.95	<0.001		47.27	<0.001
Book	3.83	0.009		57.73	<0.001
Music	9.36	<0.001		34.39	<0.001

Table 3: ANOVA results

	Watson	**LUIS**	**Rasa**
Dialogflow	**t=-5.314 p<0.001 (-)**	**t=-4.444 p<0.001 (-)**	**t=4.989 p<0.001 (-)**
Watson		t=1.508 p=0.132	t=-0.707 p=0.480
LUIS			t=0.832 p=0.405

Table 4: T-test Intent recognition results

	Watson	**LUIS**	**Rasa**	**NER_FCR**	**NER_KG**
Dialogflow	**t=6.695 p<0,001 (+)**	**t=16.622 p<0.001 (+)**	**t=18.535 p<0.001 (+)**	t=-1.234 p=0.217	**t=13.693 p<0.001 (+)**
Watson		**t=-12.627 p<0.001 (+)**	**t=-14.773 p<0.001 (+)**	**t=7.474 p<0.001 (-)**	**t=14.285 p<0.001 (+)**
LUIS			**t=-5.124 p<0.001 (+)**	**t=-16.744 p<0.001 (-)**	**t=-3.202 p=0.001 (-)**
Rasa				**t=-18.638 p<0.001 (-)**	**t=-5.390 p<0.001 (-)**
NER_FCR					**t=-8.037 p<0.001 (+)**

Table 5: T-test Entity recognition results

managed to perform very well in the ER task, obtaining a better F-score than most general-purpose platforms. However, the statistical test did not confirm that it performed better than Dialogflow. NER_KG, on the other hand, was outclassed by both Dialogflow and Watson Assistant. While this is true, it is worth noting that NER_KG was able to obtain this result by only relying on Linked data, instead of training sentences (as said in Section 4.1). This could give NER_KG the edge when developers do not possess training data at hand. The linked data-based approach also allowed it to obtain good performance in recognizing genres, where the other platforms struggled.

The answer to RQ2 is positive: all the systems that implement fuzzy matching for the ER component (Dialogflow, Watson, NER_FCR, NER_KG) performed better than those that did not implement it (LUIS, Rasa). This confirms the intuition that we presented in Section 4.1. Indeed, making sure that the system is able to tolerate spelling errors is beneficial to the overall performance of the CoRS and improves the user experience. However, we cannot conclude which fuzzy matching implementation is definitely superior: both Dialogflow and NER_FCR obtained the highest score as stated earlier. Watson Assistant also implements it, however, it performed significantly worse than the first two. Analyzing the results in detail, we can see that Watson's implementation of fuzzy matching seems limited. In fact, it is especially less tolerant of incomplete mentions. For example, given the test sentence *I like Shakespeare*, it failed to retrieve the entity *William Shakespeare*, while Dialogflow and NER_FCR did not.

Another peculiarity is that Watson seems to use fuzzy matching only on non-existent words. Given the two sentences *I like Johnny Detp* and *I like Johnny Deep*, Watson is able to recognize *Johnny Depp* correctly in the first sentence, but not in the second. NER_KG's performance in recognizing genres can be justified by the fact that it is able to exploit all aliases that are provided by Wikidata, while the other systems only worked with a single label for each entity. Genres are a particularly difficult type of entity, because they have much more variability in their surface forms, compared to titles or person names. For example, just the presence or absence of "film" or "movie" in the genre name can create issues in most systems (e.g. *I like action films* versus *I like action*). Both LUIS and Rasa lack fuzzy matching, which means that entity mentions are correctly recognized only if they match exactly their label. However, Rasa also missed some entities that had exact matching. This seems to happen especially when there are multiple entities per message.

The answer to RQ3 is negative: Rasa was able to perform just as well as all the commercial platforms in terms of Intent Recognition. On the contrary, Dialogflow performed significantly worse than Rasa and all the others. In particular, it appears that the performance of the Intent and Entity Recognition were linked in some way. By analyzing the results in further detail, we can see that most of the cases in which Dialogflow fails to recognize the *preference* intent happen when it is not able to find any entity in the sentence. In these cases, it returns the default fallback intent. This happens mostly with sentences of the form *I like [entity]*, which suggests that overfitting may have occurred.

Performing the study on three different domains allows us to explore the differences between them. These differences are especially noticeable in the Entity Recognition test. For example, we can see that the systems that the gap in ER performance between fuzzy matching and non-fuzzy matching system is more prominent in the book domain. A possible explanation for this is that users refer to book-related entities using many different surface forms, e.g. by shortening book titles or author names. Another peculiarity is that users often refer to series of books rather than single works (e.g. "Harry Potter"), which may be a potential source of confusion, as the ER component needs to disambiguate both types of references. This is an important aspect to consider when developing a natural language-based interface for a CoRS. Even though the same dialog model was originally applied for all three domains, the way the users express their preferences is still significantly different between each domain. These differences produce a noticeable effect in the overall performance of the system, which should be considered when developing a CoRS for a new domain. Understanding the users' language is key to obtain good recognition accuracy and, therefore, good user experience.

CONCLUSION

In this paper, we presented a comparison of several Natural Language Understanding systems that may be employed to create Conversational Recommender Systems. We compared their performance in the Intent and Entity Recognition tasks in three different domains: movies, books, and music. We were able to find that the platforms performed differently. For the Intent Recognition task, Watson Assistant, LUIS, and Rasa were able to achieve comparable scores. For the Entity Recognition task, Dialogflow was the best commercial platform, and performed comparably with a custom-developed Entity Recognizer (NER_FCR). It is interesting to note that some of the findings from [4] were only partly confirmed by our study. In fact, Rasa performed as well as the commercial platforms for Intent Recognition, but was instead worse in terms of Entity Recognition. Another difference is the fact that in our study Dialogflow was one of the best services for Entity Recognition, while in [4] it was one of the worst. This further confirms what emerged in that study, i.e. that the performances of NLU services are strictly dependent on the domain and, we may add, on the specific task.

As regards the limitations of our study, the first one comes from the fact that we could not include all possible platforms in the evaluation. We chose the services that we deemed the most popular. Another limitation is the size of the dataset: the number of messages used in the experiment is not very large, which could probably limit the significance of the result. As future work, we propose to collect more data, by expanding the dataset in other domains. In particular, we will make sure to increase the variability and complexity of the messages for each intent, for example, by collecting more preference messages with multiple entity mentions.

Domain	Intent	Precision	Recall	F1-score
Movie	preference	0.9987	0.9987	0.9987
	request_recommendation	1.0000	1.0000	1.0000
	show_profile	1.0000	1.0000	1.0000
	Total	0.9990	0.9990	0.9990
Music	preference	1.0000	0.9981	0.9991
	request_recommendation	1.0000	1.0000	1.0000
	show_profile	1.0000	1.0000	1.0000
	Total	1.0000	0.9987	0.9993
Books	preference	0.9977	0.9977	0.9977
	request_recommendation	1.0000	0.9930	0.9965
	show_profile	1.0000	1.0000	1.0000
	Total	0.9985	0.9969	0.9977
Overall		0.9992	0.9983	0.9988

Domain	Entity type	Precision	Recall	F1-score
Movie	item_movie	0.9752	0.9413	0.9580
	item_people	1.0000	0.9896	0.9948
	item_genre	1.0000	0.8148	0.8980
	Total	0.9854	0.9549	0.9699
Music	item_song	1.0000	1.0000	1.0000
	item_people	0.9975	0.9662	0.9816
	item_genre	0.8710	0.7297	0.7941
	Total	0.9903	0.9551	0.9724
Books	item_book	0.9889	0.9817	0.9853
	item_people	1.0000	0.9463	0.9724
	item_genre	1.0000	0.6111	0.7586
	Total	0.9929	0.9545	0.9733
Overall		0.9888	0.9549	0.9715

Table 6: Watson Conversational Agent results for IR and ER tasks

Domain	Intent	Precision	Recall	F1-score
Movie	preference	1.0000	0.9892	0.9946
	request_recommendation	0.9758	0.9806	0.9782
	show_profile	1.0000	0.9552	0.9771
	Total	0.9950	0.9852	0.9901
Music	preference	1.0000	0.9869	0.9934
	request_recommendation	0.9863	0.9863	0.9863
	show_profile	1.0000	0.9552	0.9771
	Total	0.9973	0.9839	0.9906
Books	preference	1.0000	0.9932	0.9966
	request_recommendation	0.9929	0.9789	0.9858
	show_profile	1.0000	0.9552	0.9771
	Total	0.9984	0.9861	0.9922
Overall		0.9966	0.9851	0.9908

Domain	Entity type	Precision	Recall	F1-score
Movie	item_movie	0.9934	0.9891	0.9913
	item_people	0.9965	0.9965	0.9965
	item_genre	0.9615	0.9259	0.9434
	Total	0.9935	0.9897	0.9916
Music	item_song	1.0000	1.0000	1.0000
	item_people	1.0000	0.9928	0.9964
	item_genre	1.0000	0.9459	0.9722
	Total	1.0000	0.9907	0.9953
Books	item_book	0.9818	0.9853	0.9835
	item_people	0.9932	0.9799	0.9865
	item_genre	1.0000	1.0000	1.0000
	Total	0.9863	0.9841	0.9852
Overall		0.9937	0.9886	0.9911

Table 7: Dialogflow results for IR and ER tasks

Domain	Intent	Precision	Recall	F1-score
Movie	preference	1.0000	0.9973	0.9987
	request_recommendation	0.9856	1.0000	0.9928
	show_profile	1.0000	0.9851	0.9925
	Total	0.9970	0.9970	0.9970
Music	preference	0.9981	1.0000	0.9991
	request_recommendation	1.0000	0.9932	0.9966
	show_profile	1.0000	1.0000	1.0000
	Total	0.9987	0.9987	0.9987
Books	preference	0.9910	1.0000	0.9955
	request_recommendation	1.0000	0.9789	0.9893
	show_profile	1.0000	0.9851	0.9925
	Total	0.9938	0.9938	0.9938
Overall		0.9967	0.9967	0.9967

Domain	Entity type	Precision	Recall	F1-score
Movie	item_movie	1.0000	0.8326	0.9087
	item_people	1.0000	0.9066	0.9510
	item_genre	1.0000	0.6667	0.8000
	Total	1.0000	0.8544	0.9215
Music	item_song	1.0000	0.9286	0.9630
	item_people	1.0000	0.8792	0.9357
	item_genre	0.8710	0.7297	0.7941
	Total	0.9915	0.8766	0.9306
Books	item_book	1.0000	0.8315	0.9080
	item_people	1.0000	0.7047	0.8268
	item_genre	1.0000	0.6111	0.7586
	Total	1.0000	0.7795	0.8761
Overall		0.9973	0.8424	0.9133

Table 8: LUIS results for IR and ER tasks

Domain	Intent	Precision	Recall	F1-score
Movie	preference	0.9987	0.9973	0.9980
	request_recommendation	0.9903	0.9903	0.9903
	show_profile	0.9853	1.0000	0.9926
	Total	0.9961	0.9961	0.9961
Music	preference	1.0000	1.0000	1.0000
	request_recommendation	1.0000	1.0000	1.0000
	show_profile	1.0000	1.0000	1.0000
	Total	1.0000	1.0000	1.0000
Books	preference	1.0000	0.9977	0.9989
	request_recommendation	1.0000	1.0000	1.0000
	show_profile	1.0000	1.0000	1.0000
	Total	1.0000	0.9985	0.9992
Overall		0.9983	0.9979	0.9981

Domain	Entity type	Precision	Recall	F1-score
Movie	item_movie	1.0000	0.8174	0.8995
	item_people	1.0000	0.9135	0.9548
	item_genre	1.0000	0.5926	0.7442
	Total	1.0000	0.8454	0.9162
Music	item_song	1.0000	0.9048	0.9500
	item_people	0.9945	0.8768	0.9320
	item_genre	1.0000	0.4324	0.6038
	Total	0.9956	0.8505	0.9173
Books	item_book	1.0000	0.8132	0.8970
	item_people	0.9891	0.6107	0.7552
	item_genre	1.0000	0.3889	0.5600
	Total	0.9969	0.7256	0.8399
Overall		0.9979	0.8168	0.8983

Table 9: Rasa results for IR and ER tasks

Domain	Entity type	Precision	Recall	F1-score
Movie	item_movie	0.9891	0.9848	0.9869
	item_people	1.0000	0.9896	0.9948
	item_genre	0.9643	1.0000	0.9818
	Total	0.9922	0.9871	0.9897
Music	item_song	1.0000	1.0000	1.0000
	item_people	0.9976	0.9928	0.9952
	item_genre	0.9714	0.9189	0.9444
	Total	0.9962	0.9888	0.9925
Books	item_book	1.0000	0.9853	0.9926
	item_people	1.0000	1.0000	1.0000
	item_genre	1.0000	1.0000	1.0000
	Total	1.0000	0.9909	0.9954
Overall		0.9954	0.9886	0.9920

Table 10: NER_FCR results

Domain	Entity type	Precision	Recall	F1-score
Movie	item_movie	0.9256	0.8652	0.8944
	item_people	0.9931	0.9931	0.9931
	item_genre	1.0000	0.9630	0.9811
	Total	0.9544	0.9162	0.9349
Music	item_song	0.9500	0.6786	0.7917
	item_people	0.9712	0.8961	0.9322
	item_genre	1.0000	0.8108	0.8955
	Total	0.9703	0.8561	0.9096
Books	item_book	0.8643	0.8864	0.8752
	item_people	1.0000	0.9128	0.9544
	item_genre	1.0000	0.9444	0.9714
	Total	0.9122	0.8977	0.9049
Overall		0.9479	0.8932	0.9197

Table 11: NER_KG results

REFERENCES

[1] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A Corpus for Adding Memory to Goal-Oriented Dialogue Systems. *arXiv:1704.00057 [cs]* (March 2017). http://arxiv.org/abs/1704.00057 arXiv: 1704.00057.

[2] Antoine Bordes, Y.-Lan Boureau, and Jason Weston. 2016. Learning End-to-End Goal-Oriented Dialog. *arXiv:1605.07683 [cs]* (May 2016). http://arxiv.org/abs/1605.07683 arXiv: 1605.07683.

[3] Karl Branting, James Lester, and Bradford Mott. 2004. Dialogue Management for Conversational Case-Based Reasoning. In *Advances in Case-Based Reasoning*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Peter Funk, and Pedro A. González Calero (Eds.). Vol. 3155. Springer Berlin Heidelberg, Berlin, Heidelberg, 77–90. DOI:http://dx.doi.org/10.1007/978-3-540-28631-8_7

[4] Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*. 174–185.

[5] Wanling Cai and Li Chen. 2019. Towards a Taxonomy of User Feedback Intents for Conversational Recommendations. In *RecSys*.

[6] Massimo Canonico and Luigi De Russis. 2018. A comparison and critique of natural language understanding tools. *Cloud Computing* 2018 (2018), 120.

[7] Alice Coucke, Adrien Ball, Clment Delpuech, Clment Doumouro, Sylvain Raybaud, Thibault Gisselbrecht, and Joseph Dureau. 2017. Benchmarking natural language understanding systems: Google, facebook, microsoft, amazon and snips. (2017).

[8] Kathleen Dahlgren and Edward Stabler. 1998. Natural language understanding system. (Aug. 11 1998). US Patent 5,794,050.

[9] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam,

and Jason Weston. 2015. Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. *arXiv:1511.06931 [cs]* (Nov. 2015). http://arxiv.org/abs/1511.06931 arXiv: 1511.06931.

[10] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, Ann Arbor, Michigan, 363–370. DOI: http://dx.doi.org/10.3115/1219840.1219885

[11] Milica Gavšić and Steve Young. 2011. Effective handling of dialogue state in the hidden information state POMDP-based dialogue manager. *ACM Transactions on Speech and Language Processing* 7, 3 (May 2011), 1–28. DOI: http://dx.doi.org/10.1145/1966407.1966409

[12] M. Goker and Cynthia Thompson. 2000. The adaptive place advisor: A conversational recommendation system. In *Proceedings of the 8th German Workshop on Case Based Reasoning*. Citeseer, 187–198.

[13] Andrea Iovine, Fedelucio Narducci, and Marco de Gemmis. 2019. A Dataset of Real Dialogues for Conversational Recommender Systems. In *CLiC-it 2019*. 6.

[14] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2020. A Survey on Conversational Recommender Systems. *arXiv preprint arXiv:2004.00646* (2020).

[15] Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. 2017. Understanding How People Use Natural Language to Ask for Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17*. ACM Press, Como, Italy, 229–237. DOI: http://dx.doi.org/10.1145/3109859.3109873

[16] Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. (2018), 17.

[17] Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking Natural Language Understanding Services for building Conversational Agents. *arXiv preprint arXiv:1903.05566* (2019).

[18] Tariq Mahmood and Francesco Ricci. 2009. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia - HT '09*. ACM Press, Torino, Italy, 73. DOI: http://dx.doi.org/10.1145/1557914.1557930

[19] Valentin Malykh and Vladislav Lyalin. 2018. Named Entity Recognition in Noisy Domains. In *2018 International Conference on Artificial Intelligence Applications and Innovations (IC-AIAI)*. 60–65. DOI: http://dx.doi.org/10.1109/IC-AIAI.2018.8674438 ISSN: null.

[20] Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, and others. 2016. Holographic Embeddings of Knowledge Graphs.. In *The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*. 1955–1961.

[21] Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached conversational preference elicitation: A case study in understanding movie preferences. (2019).

[22] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 41–47.

[23] S Sreelekha, Pushpak Bhattacharyya, Shishir K Jha, and D Malathi. 2016. A survey report on evolution of machine translation. *Int. J. Control Theory Appl* 9, 33 (2016), 233–240.

[24] Alessandro Suglia, Claudio Greco, Pierpaolo Basile, Giovanni Semeraro, and Annalina Caputo. 2017. An Automatic Procedure for Generating Datasets for Conversational Recommender Systems. (2017), 2.

[25] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[26] Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The Dialog State Tracking Challenge Series: A Review. *Dialogue & Discourse* 7, 3 (April 2016), 4–33. http://dad.uni-bielefeld.de/index.php/dad/article/view/3685

[27] Caroline Wisniewski, Clment Delpuech, David Leroy, Franois Pivan, and Joseph Dureau. 2017. Benchmarking Natural Language Understanding Systems. (2017). https://snips.ai/content/sdk-benchmark-visualisation/