Evaluating Natural Language Understanding Services for Conversational Question Answering Systems

Daniel Braun **Adrian Hernandez Mendez** **Florian Matthes**
Technical University of Munich
Department of Informatics
{daniel.braun,adrian.hernandez,matthes}
@tum.de

Manfred Langen
Siemens AG
Corporate Technology
manfred.langen
@siemens.com

Abstract

Conversational interfaces recently gained a lot of attention. One of the reasons for the current hype is the fact that chatbots (one particularly popular form of conversational interfaces) nowadays can be created without any programming knowledge, thanks to different toolkits and so-called Natural Language Understanding (NLU) services. While these NLU services are already widely used in both, industry and science, so far, they have not been analysed systematically. In this paper, we present a method to evaluate the classification performance of NLU services. Moreover, we present two new corpora, one consisting of annotated questions and one consisting of annotated questions with the corresponding answers. Based on these corpora, we conduct an evaluation of some of the most popular NLU services. Thereby we want to enable both, researchers and companies to make more educated decisions about which service they should use.

1 Introduction

Long before the terms *conversational interface* or *chatbot* were coined, Turing (1950) described them as the ultimate test for artificial intelligence. Despite their long history, there is a recent hype about chatbots in both, the scientific community (cf. e.g. Ferrara et al. (2016)) and industry (Gartner, 2016). While there are many related reasons for this development, we think that three key changes were particularly important:

- Rise of universal chat platforms (like Telegram, Facebook Messenger, Slack, etc.)
- Advances in machine learning (ML)
- Natural Language Understanding (NLU) as a service

In this paper, we focus on the latter. As we will show in Section 2, NLU services are already used by a number of researchers for building conversational interfaces. However, due to the lack of a systematic evaluation of theses services, the decision why one services was prefered over another, is usually not well justified. With this paper, we want to bridge this gap and enable both, researchers and companies, to make more educated decisions about which service they should use. We describe the functioning of NLU services and their role within the general architecture of chatbots. We explain, how NLU services can be evaluated and conduct an evaluation, based on two different corpora consisting of nearly 500 annotated questions, of the most popular services.

2 Related Work

Recent publications have discussed the usage of NLU services in different domains and for different purposes, e.g. question answering for localized search (McTear et al., 2016), form-driven dialogue systems (Stoyanchev et al., 2016), dialogue management (Schnelle-Walka et al., 2016), and the internet of things (Kar and Haldar, 2016).

However, none of these publications explicitly discuss, why they choose one particular NLU service over another and how this decision may have influenced the performance of their system and hence their results. Moreover, to the best of our knowledge, so far there exists no systematic evaluation of a particular NLU service, let alone a comparison of multiple services.

Dale (2015) lists five NLP cloud services and describes their capabilities, but without conducting an evaluation. In the domain of spoken dialog

Proceedings of the SIGDIAL 2017 Conference, pages 174–185,
Saarbrücken, Germany, 15-17 August 2017.

systems, similar evaluations have been conducted for automatic speech recognizer services, e.g. by Twiefel et al. (2014) and Morbini et al. (2013).

Speaking about chatbots in general, Shawar and Atwell (2007) present an approach to conduct end-to-end evaluations, however, they do not take into account the single elements of a system. Resnik and Lin (2010) provide a good overview and evaluation of Natural Language Processing (NLP) systems in general. Many of the principals they apply for their evaluation (e.g. inter-annotator agreement and partitioning of data) play an important role in our evaluation too. A comprehensive and extensive survey of question answering technologies was presented by Kolomiyets and Moens (2011). However, there has been a lot of progress since 2011, including the here presented NLU services.

One of our two corpora was labelled using Amazon Mechanical Turk (AMT, cf. Section 5.2), while there have been long discussions about whether or not AMT can replace the work of experts for labelling linguistic data, the recent consensus is that, given enough annotators, crowd-sourced labels from AMT are as reliable as expert data. (Snow et al., 2008; Munro et al., 2010; Callison-Burch, 2009)

3 Chatbot Architecture

In order to understand the role of NLU services for chatbots, one first has to look at the general architecture of chatbots. While there exist different documented chatbot architectures for concrete use cases, no universal model of how a chatbot should be designed has emerged yet. Our proposal for a universal chatbot architecture is shown in Figure 1. It consists of three main parts: Request Interpretation, Response Retrieval and Message Generation. The Message Generation follows the classical Natural Language Generation (NLG) pipeline described by Reiter and Dale (2000). In the context of Request Interpretation, a "request" is not necessarily a question, but can also be any user input like "My name is John". Equally, a "response" to this input could e.g. be "What a nice name".

4 NLU Services

The general goal of NLU services is the extraction of structured, semantic information from unstructured natural language input, e.g. chat messages. They mainly do this by attaching user-defined labels to messages or parts of messages. At the time of writing, among the most popular NLU services are:

- LUIS[1]
- Watson Conversation[2]
- API.ai[3]
- wit.ai[4]
- Amazon Lex[5]

Moreover, there is a popular open source alternative which is called RASA[6]. RASA offers the same functionality, while lacking the advantages of cloud-based solutions (managed hosting, scalability, etc). On the other hand, it offers the typical advantages of self-hosted open source software (adaptability, data control, etc).

Table 1 shows a comparison of the basic functionality offered by the different services. All of them, except for Amazon Lex, share the same basic concept: Based on example data, the user can train a classifier to classify so-called intents (which represent the intent of the whole message and are not bound to a certain position within the message) and entities (which can consist of a single or multiple characters).

Service	**Intents**	**Entities**	**Batch import**
LUIS	+	+	+
Watson	+	+	+
API.ai	+	+	+
wit.ai	+	+	O
Lex	+	O	-
RASA	+	+	+

Table 1: Comparison basic functionality of NLU services

Figure 2 shows a labelled sentence in the LUIS web interface. The intent of this sentence was classified as FindConnection, with a confidence of 97%. The labelled entities are: (next, Criterion), (train, Vehicle), (Universität, StationStart), (Max-Weber-Platz, StationDest). Amazon Lex shares

[1] https://www.luis.ai
[2] https://www.ibm.com/watson/developercloud/conversation.html
[3] https://www.api.ai
[4] https://www.wit.ai
[5] https://aws.amazon.com/lex
[6] https://www.rasa.ai

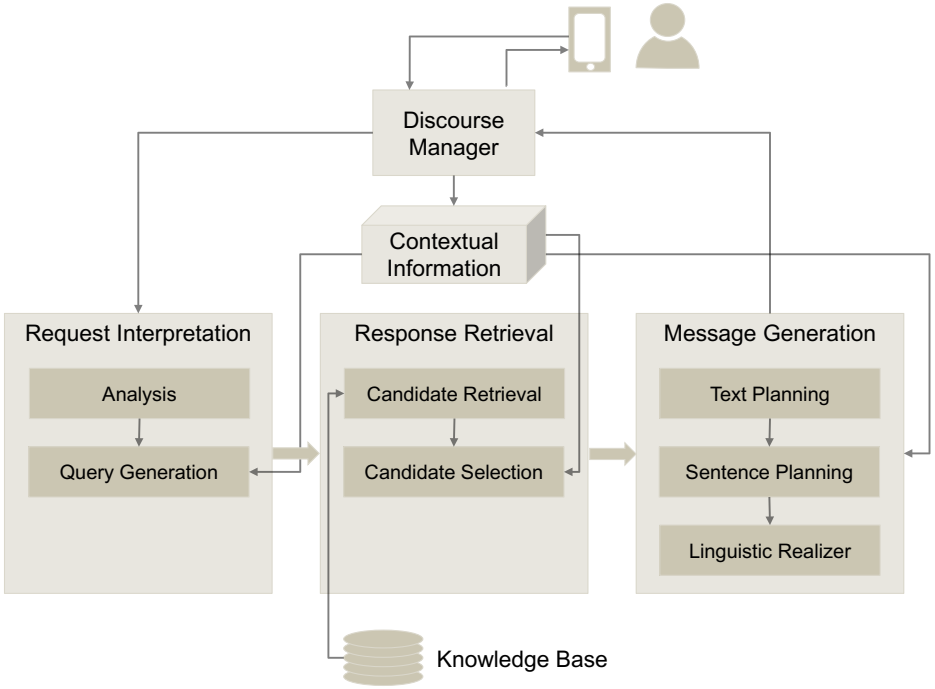

Figure 1: General Architecture for Chatbots

the concept of intents with the other services, but instead of entities, Lex is using so-called slots, which are not trained by concrete examples, but example patterns like "When is the {Criterion} {Vehicle} to {StationDest}". Moreover, all services, except for Amazon Lex, also offer an export and import functionality which uses a json-format to export and import the training data. While wit.ai offers this functionality, as of today, it only works reliably for creating backups and restoring them, but not importing new data[7].

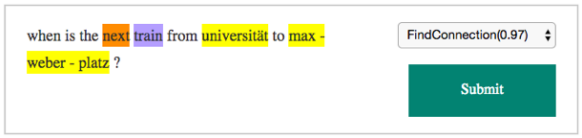

Figure 2: Labelled sentence with intent and entities in Microsoft LUIS

When it comes to the core of the services, the machine learning algorithms and the data on which they are initially trained, all services are very secretive. None of them gives specific information about the used technologies and datasets. The exception in this case is, of course, RASA, which can either use MITIE (Geyer et al., 2016) or spaCy (Choi et al., 2015) as ML backend.

5 Data Corpus

Our evaluation is based on two very different data corpora. The Chatbot Corpus (cf. Section 5.1) is based on questions gathered by a *Telegram* chatbot in production use, answering questions about public transport connections. The StackExchange Corpus (cf. Section 5.2) is based on data from two StackExchange[8] platforms: *ask ubuntu*[9] and *Web Applications*[10]. Both corpora are available on GitHub under the Creative Commons CC BY-SA 3.0 license[11]: https://github.com/sebischair/NLU-Evaluation-Corpora.

5.1 Chatbot Corpus

The Chatbot Corpus consists of 206 questions, which were manually labelled by the authors. There are two different intents (Departure Time,

[7] cf. e.g. https://github.com/wit-ai/wit/issues?q=is%3Aopen+is%3Aissue+label%3Aimport

[8] https://www.stackexchange.com

[9] https://www.askubuntu.com

[10] https://webapps.stackexchange.com

[11] https://creativecommons.org/licenses/by-sa/3.0/

Find Connection) in the corpus and five different entity types (StationStart, StationDest, Criterion, Vehicle, Line). The general language of the questions was English, however, mixed with German street and station names. Example entries from the corpus can be found in Appendix A.1. For the evaluation, the corpus was split into a training dataset with 100 entries and a test dataset with 106 entries.

43% of the questions in the training dataset belong to the intent Departure Time and 57% to Find Connection. The distribution for the test dataset is 33% (Departure Time) and 67% (Find Connection). Table 2 shows how the different entity types are distributed among the two datasets. While some entity types occur very often, like StationStart, some occur very rarely, especially Line. We do this differentiation to evaluate, if some services handle very common, or very rare, entity types better than others.

While in this corpus, there are more tagged entities in the training dataset than in the test dataset, it is the other way round in the other corpus, which will be introduced in the next section. Although one might expect that this leads to better results, the evaluation in Section 7 shows that this is not necessarily the case.

Entity Type	**training**	**test**	Σ
StationStart	91	102	193
StationDest	57	71	128
Criterion	48	34	82
Vehicle	50	35	85
Line	4	2	6
Σ	250	244	494

Table 2: Entity types within the chatbot corpus

5.2 StackExchange Corpus

For the generation of the StackExchange corpus, we used the StackExchange Data Explorer[12]. We choose the most popular questions (i.e. questions with the highest scores and most views), from the two StackExchange platforms *ask ubuntu* and *Web Applications*, because they are likely to have a better linguistic quality and a higher relevance, compared to less popular questions. Additionally, we used only questions with an accepted, i.e. correct, answer. Although we did not use the answers in our evaluation, we included them in our corpus, in order to create a corpus that is not only useful for this particular evaluation, but also for research on question answering in general. In this way, we gathered 290 questions and answers in total, 100 from *Web Applications* and 190 from *ask ubuntu*.

The corpus was labelled with intents and entities using Amazon Mechanical Turk (AMT). Each question was labelled by five different workers, summing up to nearly 1,500 datapoints.

For each platform, we created a list of candidates for **intents**, which were extracted from the labels (i.e. tags) assigned to the questions by StackExchange users. For each question, the AMT workers were asked to chose one of these intents or "None", if they think no candidate is fitting.

For *ask ubuntu*, the possible intents were: "Make Update", "Setup Printer", "Shutdown Computer", and "Software Recommendation".

For *Web Applications*, the candidates were: "Change Password", "Delete Account", "Download Video", "Export Data", "Filter Spam", "Find Alternative", and "Sync Accounts".

Similarly, a set of **entity type** candidates were given. By marking parts of the questions with the mouse, workers could assign these entity types to words (or characters) within the question. For *Web Applications* the possible entity types were: "WebService", "OperatingSystem" and "Browser". For *ask ubuntu*, they were: "SoftwareName", "Printer", and "UbuntuVersion".

Moreover, workers were asked to state how confident they are in their assessment: very confident, somewhat confident, undecided, somewhat unconfident, or very unconfident.

For the generation of the annotated, final corpus, only submissions with a confidence level of "undecided" or higher were taken into account. A label, no matter if intent or entity, was only added to the corpus if the inter-annotator agreement among those confident annotators was 60% or higher. If no intent could be found for a question, satisfying these criteria, this question was not added to the corpus. The final corpus was also checked for false positives by two experts, but non were found. Therefore the final corpus consists of 251 entries, 162 from *ask ubuntu* and 89 from *Web Applications*. Example entries from the corpus are shown in Appendix A.2.

For the evaluation, we also split this corpus.

[12] https://data.stackexchange.com

Four datasets were separated, one for training and one for testing, for each platform. The distribution of intents among these datasets is shown in Table 3, the distribution of entity types is shown in Table 4. Again, we do this differentiation to compare the classification results for frequently and rarely occurring intents and entity types.

Intent	**training**	**test**	Σ
ChangePassword	2	6	8
DeleteAccount	7	10	17
DownloadVideo	1	0	1
ExportData	2	3	5
FilterSpam	6	14	20
FindAlternative	7	16	23
SyncAccounts	3	6	9
None	2	4	6
Σ	30	54	84

(a) *Web Applications* datasets

Intent	**training**	**test**	Σ
MakeUpdate	10	37	47
SetupPrinter	10	13	23
ShutdownComputer	13	14	27
S.Recommendation	17	40	57
None	3	5	8
Σ	53	109	162

(b) *ask ubuntu* datasets

Table 3: Intents within StackExchange corpus

dataset	Entity Type	**training**	**test**	Σ
web apps	WebService	33	64	97
	OS	1	0	1
	Browser	1	0	1
	Σ	35	64	99
ubuntu	Printer	8	12	20
	Software	3	4	7
	Version	24	78	102
	Σ	35	94	129

Table 4: Entity types within the StackExchange corpus

6 Experimental Design

In order to compare the performance of the different NLU services, we used the corpora described in Section 5. We used the respective training datasets to train the NLU services LUIS, Watson Conversation, API.ai, and RASA. Amazon Lex was not included in this comparison because, as mentioned in Section 4, it does not offer a batch import functionality, which is crucial in order to effectively train all services with the exact same data. For the same reason, wit.ai was also excluded from the experiment. While it does offer an import option, currently, it only works reliable for data which was created through the wit.ai webinterface and not altered, or even created, manually.

Afterwards, the test datasets were sent to the NLU services and the labels created by the services were compared against our human created gold standard. For training, we used the batch import interfaces, offered by all compared services, in this way it was not only possible to train all different services relatively fast, despite many hundred individual labels, it also guaranteed, that all services are fed with exactly the same data. Since the data format differs from service to service, we used a Python script to automatically convert the training datasets from the format shown in the Appendix to the respective data format of the services. For retrieving the results for the test datasets from the NLU services, their respective REST-APIs were used.

In order to evaluate the results, we calculated true positives, false positives, and false negatives, based on exact matches. Based on this data, we computed precision and recall as well as F-score for single intents, entity types, and corpora, as well as overall results. We will say one service is better than another if it has a higher F-score.

6.1 Hypotheses

Before the conduction of the experiment, we had three main hypotheses:

1. **The performance varies between services**: Although it might sound obvious, it is worth mentioning that one of the reasons for this evaluation is the fact that we think, there is a difference between the compared NLU services. Despite their very similar concepts and "look and feel", we expect differences when it comes to annotation quality (i.e. F-scores), which should be taken into account when deciding for one or another service.

2. **The commercial products will (overall) perform better**:

The initial language model of RASA, which comes with MITIE, is about 300 MB of data. The commercial services, on the other hand, are fed with data by hundreds, if not thousands, of users every day. We, therefore, assume, that the commercial products will perform better in the evaluation, especially when the training data is sparse.

3. **The quality of the labels is influenced by the domain**:
   We assume that, depending on the used algorithms and models, individual services will perform differently in different domains. Therefore, we think it is not unlikely that a service which performs well on the more technical corpus from StackExchange will perform considerably worse on the chatbot corpus, which has a focus on spatial and time data, and vice versa.

6.2 Limitations

One important limitation of this evaluation is the fact that the results will not be representative for other domains. On the opposite, as already mentioned in Hypothesis 3, we do believe that there are important differences in performance between different domains. Therefore our final conclusion can not be that one service is absolutely better than the others, but rather that on the given corpus, one service performed better than the others. However, we believe that the here presented approach will help developers to conduct evaluations of NLU services for their domain and thus empower them to make better-informed decisions.

With regard to the used corpora, we made an effort to make them as naturally as possible by using only real data from real users. However, when analysing the results, one should keep in mind that the Chatbot Corpus consists of questions which were asked by users, which were aware of communicating with a chatbot. It is, therefore, conceivable that they formulated their questions in a way which they expect to be more understandable for a chatbot.

Finally, NLU services, like all other services, can change over time (and hopefully improve). While it is easy to track these changes for locally installed software, changes on cloud-based services may happen without any notice to the user. Conducting the very same experiment, described in this paper, in six months time, might, therefore, lead to different results. This evaluation can therefore only be a snapshot of the current state of the compared services. While this might decrease the reproducibility of our experiment, it is also a good argument for a formalized, repeatable evaluation process, as we describe it in this paper.

7 Evaluation

The detailed results of the evaluation, broken down on single intents, entity types, corpora, and overall, are shown in Table 5 to 8. Each table shows the result from a different NLU service. Within the tables, each row represents one particular entity type or intent.

For each row, the corpus, type (intent/entity), and true positives, false negatives, and false positives are given. From these values, precision, recall, and F-score have been calculated. The entity types and intents are also sorted by the corpus they appear in. For each corpus, there is a summary row, which shows precision, recall, and F-score for the whole corpus. At the bottom of each table, there is also an overall summary.

From a high-level perspective, LUIS performed best with an F-score of 0.916, followed by RASA (0.821), Watson Conversation (0.752), and API.ai (0.687). LUIS also performed best on each individual dataset: chatbot, web apps, and ask ubuntu. Similarly, API.ai performed worst on every dataset, while the second place changes between RASA and Watson Conversation (cf. Figure 3).

Based on this data, the second hypothesis can be rejected. Although the best performance was indeed shown by a commercial product, RASA easily competes with the other commercial products.

The first hypothesis is supported by our findings. We can see a difference between the services, with the F-score of LUIS being nearly 0.3 higher than the F-score of API.ai. However, a conducted two-way ANOVA analysis with the F-score as dependent variable and the NLU service and the entity type/intent as fixed factors does not show a significance at the level of $p < 0.05$ ($p = 0.234, df = 3$). An even larger corpus might be necessary to get quantitatively more robust results.

With regard to the third hypothesis, the picture is less clear. Although we can see a clear influence of the domain on the F-score within each service, the ranking between different services is not

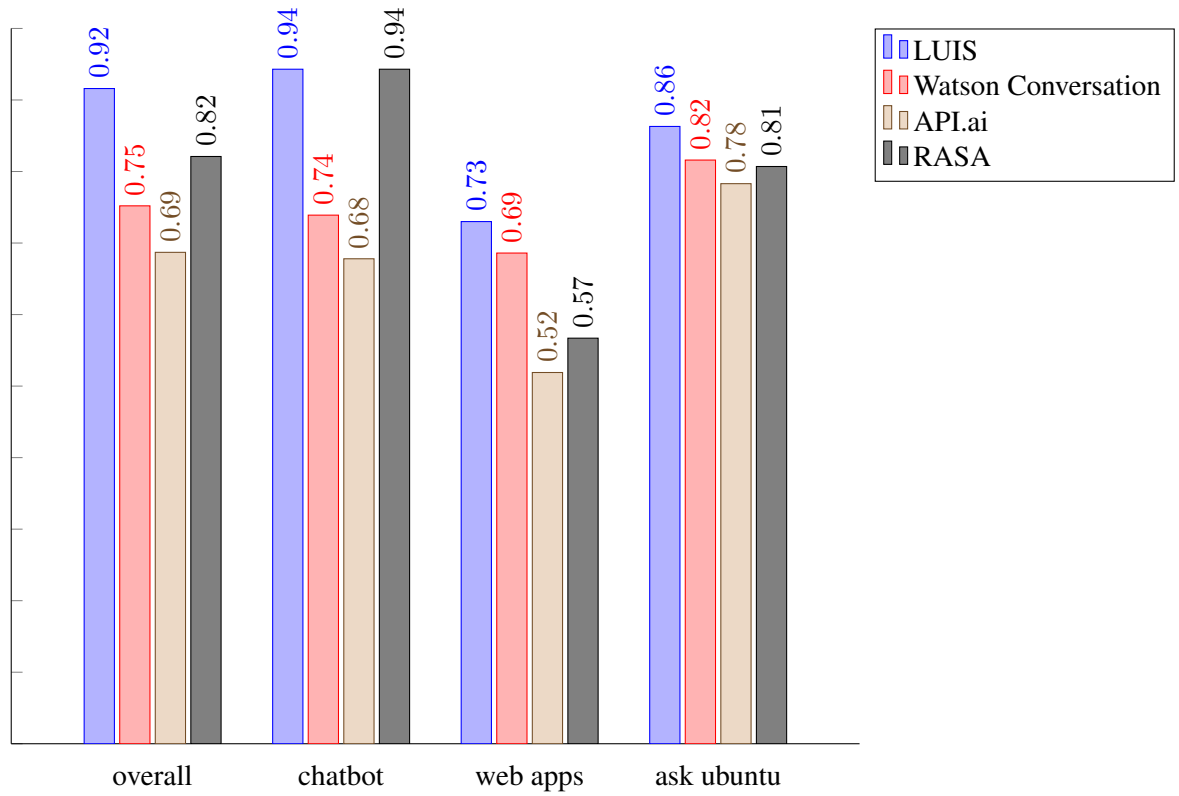

Figure 3: F-scores for the different NLU services, grouped by corpus

much influenced. LUIS always performs best, independent from the domain, API.ai always worst, also independent from the domain, merely the second and third place changes. Therefore, although the domain influences the results, it is not clear whether or not it should also influence the decision which service should be used.

On a more detailed level, we also see differences between entities and intents. Especially API.ai seems to have big troubles identifying entities. On the web apps corpus, for example, API.ai did not identify a single occurrence of the entity type WebService, which occurred 64 times in the dataset. If we calculate the F-score for this dataset only based on the intents, it would increase from 0.519 to 0.803. The overall results of API.ai were therefore heavily influenced by its shortcomings regarding entity detection.

If we look at intents and entity types with sparse training data, like Line, ChangePassword, and ExportData, other than we expected, we do not see a significantly better performance of commercial services.

8 Conclusion

The evaluation of the NLU services LUIS, Watson Conversation, API.ai, and RASA, based on the two corpora we presented in Section 5, has shown that the quality of the annotations differs between the different services. Before using an NLU service, no matter if for commercial or scientific purposes, one should therefore compare the different services with domain specific data.

For our two corpora, LUIS showed the best results, however, the open source alternative RASA could achieve similar results. Given the advantages of open source solutions (mainly adaptability), it might well be possible to achieve an even better results with RASA, after some customization.

With regard to absolute numbers, it is difficult to decide whether an F-score of 0.916 or 0.821 is satisfactory for productive use within a conversational question answering system. This decision also depends strongly on the concrete use case. We, therefore, focused on relative comparisons in our evaluation and leave this decision to future users.

corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	34	1	1	0.971	0.971	0.971
	FindConnection	Intent	70	1	1	0.986	0.986	0.986
	Criterion	Entity	34	0	0	1	1	1
	Line	Entity	0	2	0		0	
	StationDest	Entity	65	6	3	0.956	0.915	0.935
	StationStart	Entity	90	17	5	0.947	0.841	0.891
	Vehicle	Entity	33	2	0	1	0.943	0.971
	Σ		326	29	10	0.970	0.918	0.943
web apps	ChangePassword	Intent	3	3	0	1	0.5	0.667
	DeleteAccount	Intent	8	2	0	1	0.8	0.889
	DownloadVideo	Intent	0	0	0			
	ExportData	Intent	3	0	1	0.75	1	0.857
	FilterSpam	Intent	12	2	0	1	0.857	0.923
	FindAlternative	Intent	14	2	2	0.875	0.875	0.875
	None	Intent	3	1	8	0.273	0.75	0.4
	SyncAccounts	Intent	5	1	0	1	0.833	0.909
	WebService	Entity	29	30	5	0.853	0.492	0.624
	Σ		77	41	16	0.828	0.653	0.73
ask ubuntu	MakeUpdate	Intent	36	1	4	0.900	0.973	0.935
	SetupPrinter	Intent	12	1	2	0.857	0.923	0.889
	ShutdownComputer	Intent	14	0	0	1	1	1
	SRecommendation	Intent	36	4	5	0.878	0.9	0.889
	None	Intent	0	5	0		0	
	SoftwareName	Entity	0	4	0		0	
	Printer	Entity	5	7	0	1	0.417	0.589
	UbuntuVersion	Entity	67	10	11	0.859	0.87	0.864
	Σ		170	32	22	0.885	0.842	0.863
overall			820	102	48	0.945	0.889	0.916

Table 5: Results LUIS

corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	33	2	1	0.971	0.943	0.957
	FindConnection	Intent	70	1	2	0.972	0.986	0.979
	Criterion	Entity	34	0	0	1	1	1
	Line	Entity	1	1	0	1	0.5	0.667
	StationDest	Entity	42	29	75	0.359	0.592	0.447
	StationStart	Entity	65	37	50	0.565	0.637	0.599
	Vehicle	Entity	35	0	0	1	1	1
	Σ		280	70	128	0.686	0.8	0.739
web apps	ChangePassword	Intent	5	1	0	1	0.833	0.909
	DeleteAccount	Intent	9	1	3	0.750	0.9	0.818
	DownloadVideo	Intent	0	0	1	0		
	ExportData	Intent	2	1	2	0.500	0.667	0.572
	FilterSpam	Intent	13	1	2	0.867	0.929	0.897
	FindAlternative	Intent	15	1	1	0.938	0.938	0.938
	None	Intent	0	4	1	0	0	
	SyncAccounts	Intent	5	1	0	1	0.833	0.909
	WebService	Entity	23	41	5	0.821	0.359	0.5
	Σ		72	51	15	0.828	0.585	0.686
ask ubuntu	MakeUpdate	Intent	37	0	4	0.902	1	0.948
	SetupPrinter	Intent	13	0	1	0.929	1	0.963
	ShutdownComputer	Intent	14	0	0	1	1	1
	SRecommendation	Intent	35	5	3	0.921	0.875	0.897
	None	Intent	1	4	1	0.500	0.2	0.286
	SoftwareName	Entity	0	4	0		0	
	Printer	Entity	0	12	0		0	
	UbuntuVersion	Entity	51	7	27	0.654	0.879	0.75
	Σ		151	32	36	0.807	0.825	0.816
overall			503	153	179	0.738	0.767	0.752

Table 6: Results Watson Conversation

corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	35	0	4	0.897	1	0.946
	FindConnection	Intent	60	11	0	1	0.845	0.916
	Criterion	Entity	31	3	0	1	0.912	0.954
	Line	Entity	1	1	0	1	0.5	0.667
	StationDest	Entity	0	71	0		0	
	StationStart	Entity	28	79	4	0.875	0.262	0.403
	Vehicle	Entity	34	1	5	0.872	0.971	0.919
	Σ		189	166	13	0.936	0.532	0.678
web apps	ChangePassword	Intent	4	2	1	0.800	0.667	0.727
	DeleteAccount	Intent	10	0	2	0.833	1	0.909
	DownloadVideo	Intent	0	0	0			
	ExportData	Intent	1	2	2	0.333	0.333	0.333
	FilterSpam	Intent	10	4	3	0.769	0.714	0.74
	FindAlternative	Intent	16	0	2	0.889	1	0.941
	None	Intent	2	2	1	0.667	0.5	0.572
	SyncAccounts	Intent	4	2	0	1	0.667	0.8
	WebService	Entity	0	64	0		0	
	Σ		47	76	11	0.810	0.382	0.519
ask ubuntu	MakeUpdate	Intent	36	1	3	0.923	0.973	0.947
	SetupPrinter	Intent	13	0	1	0.929	1	0.963
	ShutdownComputer	Intent	14	0	2	0.875	1	0.933
	SRecommendation	Intent	28	12	2	0.933	0.7	0.8
	None	Intent	2	3	8	0.200	0.4	0.267
	SoftwareName	Entity	0	4	0		0	
	Printer	Entity	0	12	0		0	
	UbuntuVersion	Entity	48	30	0	1	0.615	0.762
	Σ		141	46	32	0.815	0.754	0.783
overall			377	288	56	0.871	0.567	0.687

Table 7: Results API.ai

corpus	entity type / intent	type	true +	false -	false +	precision	recall	F-score
chatbot	DepartureTime	Intent	34	1	1	0.971	0.971	0.971
	FindConnection	Intent	70	1	1	0.986	0.986	0.986
	Criterion	Entity	34	0	0	1	1	1
	Line	Entity	0	2	0		0	
	StationDest	Entity	65	6	3	0.956	0.915	0.935
	StationStart	Entity	90	17	5	0.947	0.841	0.891
	Vehicle	Entity	33	2	0	1	0.943	0.971
	Σ		326	29	10	0.970	0.918	0.943
web apps	ChangePassword	Intent	4	2	0	1	0.667	0.8
	DeleteAccount	Intent	9	1	5	0.643	0.9	0.75
	DownloadVideo	Intent	0	0	1	0		
	ExportData	Intent	0	3	0		0	
	FilterSpam	Intent	13	1	0	1	0.929	0.963
	FindAlternative	Intent	15	1	8	0.652	0.938	0.769
	None	Intent	0	4	1	0	0	
	SyncAccounts	Intent	3	3	0	1	0.5	0.667
	WebService	Entity	45	19	87	0.341	0.703	0.459
	Σ		89	34	102	0.466	0.724	0.567
ask ubuntu	MakeUpdate	Intent	34	3	2	0.944	0.919	0.931
	SetupPrinter	Intent	13	0	2	0.867	1	0.929
	ShutdownComputer	Intent	14	0	6	0.700	1	0.824
	SRecommendation	Intent	33	7	4	0.892	0.825	0.857
	None	Intent	0	5	1	0	0	
	SoftwareName	Entity	0	4	11	0	0	
	Printer	Entity	8	4	11	0.421	0.667	0.516
	UbuntuVersion	Entity	65	13	7	0.903	0.833	0.867
	Σ		167	36	44	0.791	0.823	0.807
overall			582	99	156	0.789	0.855	0.821

Table 8: Results RASA

References

Chris Callison-Burch. 2009. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, pages 286–295.

Jinho D. Choi, Joel R. Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using A web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 387–396. http://aclweb.org/anthology/P/P15/P15-1038.pdf.

Robert Dale. 2015. Nlp meets the cloud. *Natural Language Engineering* 21(4):653–659. https://doi.org/10.1017/S1351324915000200.

Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. 2016. The rise of social bots. *Communications of the ACM* 59(7):96–104.

Gartner. 2016. Hype cycle for emerging technologies, 2016. Technical report. http://www.gartner.com/document/3383817.

Kelly Geyer, Kara Greenfield, Alyssa Mensch, and Olga Simek. 2016. Named entity recognition in 140 characters or less. In *6th Workshop on Making Sense of Microposts (#Microposts2016)*. pages 78–79.

Rohan Kar and Rishin Haldar. 2016. Applying chatbots to the internet of things: Opportunities and architectural elements. *arXiv preprint arXiv:1611.03799* .

Oleksandr Kolomiyets and Marie-Francine Moens. 2011. A survey on question answering technology from an information retrieval perspective. *Inf. Sci.* 181(24):5412–5434. https://doi.org/10.1016/j.ins.2011.07.047.

Michael McTear, Zoraida Callejas, and David Griol. 2016. *The Conversational Interface: Talking to Smart Devices*, Springer International Publishing, Cham, chapter Implementing Spoken Language Understanding, pages 187–208.

Fabrizio Morbini, Kartik Audhkhasi, Kenji Sagae, Ron Artstein, Dogan Can, Panayiotis Georgiou, Shri Narayanan, Anton Leuski, and David Traum. 2013. Which asr should i choose for my dialogue system. In *Proceedings of the 14th annual SIGdial Meeting on Discourse and Dialogue*. pages 394–403.

Robert Munro, Steven Bethard, Victor Kuperman, Vicky Tzuyin Lai, Robin Melnick, Christopher Potts, Tyler Schnoebelen, and Harry Tily. 2010. Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk*. Association for Computational Linguistics, pages 122–130.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.

Philip Resnik and Jimmy Lin. 2010. Evaluation of nlp systems. *The handbook of computational linguistics and natural language processing* 57.

Dirk Schnelle-Walka, Stefan Radomski, Benjamin Milde, Chris Biemann, and Max Mühlhäuser. 2016. Nlu vs. dialog management: To whom am i speaking? In *Joint Workshop on Smart Connected and Wearable Things (SCWT'2016), co-located with IUI*. https://doi.org/10.13140/RG.2.1.1928.4247.

Bayan Abu Shawar and Eric Atwell. 2007. Different measurements metrics to evaluate a chatbot system. In *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL-HLT-Dialog '07, pages 89–96. http://dl.acm.org/citation.cfm?id=1556328.1556341.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '08, pages 254–263. http://dl.acm.org/citation.cfm?id=1613715.1613751.

Svetlana Stoyanchev, Pierre Lison, and Srinivas Bangalore. 2016. Rapid prototyping of form-driven dialogue systems using an open-source framework. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, Los Angeles, pages 216–219. http://www.aclweb.org/anthology/W16-3626.

Alan M Turing. 1950. Computing machinery and intelligence. *Mind* 59(236):433–460.

Johannes Twiefel, Timo Baumann, Stefan Heinrich, and Stefan Wermter. 2014. Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing. In *AAAI*. pages 1529–1536.

A Supplemental Material

A.1 Examples Chatbot Corpus


```
{
  "text": "what is the cheapest
     ↪ connection between
     ↪ quiddestraße and
     ↪ hauptbahnhof?",
  "intent": "FindConnection",
  "entities": [
    {
      "entity": "Criterion",
      "start": 3,
      "stop": 3
    },
    {
      "entity": "StationStart",
      "start": 6,
      "stop": 6
    },
    {
      "entity": "StationDest",
      "start": 8,
      "stop": 8
    }
  ]
},
{
  "text": "when is the next u6
     ↪ leaving from garching?",
  "intent": "DepartureTime",
  "entities": [
    {
      "entity": "Line",
      "start": 4,
      "stop": 4
    },
    {
      "entity": "StationStart",
      "start": 7,
      "stop": 7
    }
  ]
}
```


A.2 Examples StackExchange Corpus

A.2.1 Web Applications Dataset


```
{
  "text": "How can I delete my
     ↪ Twitter account?",
  "url": "http://
     ↪ webapps.stackexchange.com
     ↪ /questions/57/how-can-i-
     ↪ delete-my-twitter-account
     ↪ ",
  "author": "Jared Harley",
  "answer": {
    "text": "[...]",
    "author": "Ken Pespisa"
  },
  "intent": "Delete Account",
  "entities": [
    {
      "text": "Twitter",
      "stop": 5,
      "start": 5,
      "entity": "WebService"
    }
  ]
},
{
  "text": "Is it possible to
     ↪ export my data from
     ↪ Trello to back it up?",
  "url": "http://
     ↪ webapps.stackexchange.com
     ↪ /questions/18975/is-it-
     ↪ possible-to-export-my-
     ↪ data-from-trello-to-back-
     ↪ it-up",
  "author": "Clare Macrae",
  "answer": {
    "text": "[...]",
    "author": "Daniel LeCheminant
       ↪ "
  },
  "intent": "Export Data",
  "entities": [
    {
      "text": "Trello",
      "stop": 8,
      "start": 8,
      "entity": "WebService"
    }
  ]
}
```


A.2.2 Ask Ubuntu Dataset


```
{
  "text": "How do I install the
     ↪ HP F4280 printer?",
  "url": "http://askubuntu.com/
```

```
      ↪ questions/24073/how-do-i-
      ↪ install-the-hp-f4280-
      ↪ printer",
    "author": "ok comp",
    "answer": {
      "text": "[...]",
      "author": "nejode"
    },
    "intent": "Setup Printer",
    "entities": [
      {
        "text": "HP F4280",
        "stop": 6,
        "start": 5,
        "entity": "Printer"
      }
    ]
  },
  {
    "text": "What is a good MongoDB
      ↪  GUI client?",
    "url": "http://askubuntu.com/
      ↪ questions/196136/what-is-
      ↪ a-good-mongodb-gui-client
      ↪ ",
    "author": "Eyal",
    "answer": {
      "text": "[...]",
      "author": "Eyal"
    },
    "intent": "Software
      ↪ Recommendation",
    "entities": [
      {
        "text": "MongoDB",
        "stop": 4,
        "start": 4,
        "entity": "SoftwareName"
      }
    ]
  }
```