A Comparison and Critique of Natural Language Understanding Tools

A Comparison and Critique of Natural Language Understanding Tools

Massimo Canonico

Department of Science and Innovation Technology
University of Piemonte Orientale
Italy
Email: massimo.canonico@uniupo.it

Luigi De Russis

Department of Control and Computer Engineering
Politecnico di Torino
Italy
Email: luigi.derussis@polito.it

***Abstract*—In the last 10 years, various cloud platforms enabled developers to easily create applications able to understand, with some limitations, natural languages. Nowadays, such cloud platforms for natural language understanding (NLU) are widely used, thanks to the rise of multiple chat services and conversational assistants on our mobile devices. This paper compares and analyses the main cloud-based NLU platforms, both from a descriptive and from a performance-based point of view. For the descriptive analysis, a taxonomy is proposed and six cloud platforms are analyzed. The performance evaluation, instead, compares three of these platforms, highlighting strengths and weaknesses of the different NLU tools.**

***Keywords*–Natural Language Understanding; Cloud Platform; Comparison; NLU Taxonomy.**

I. Introduction

Over the last 20 years, computational linguistics has grown very fast both in scientific research and practical technology that are being incorporated into customer products (for example, in applications such as Apple's Siri [1] or in hardware components such as Google Home [2]). This has been possible thanks to four key factors: (i) a vast increase in computing power, (ii) the huge amount of linguistic data available, (iii) the improvement of Machine Learning, and (iv) a better understanding of the structure of human language.

Natural Language Understanding (NLU) is a subfield of computer science concerned with the usage of computational techniques to learn, understand, and produce human language content [3]. NLU can have multiple purposes: from aiding human-human communication (e.g., Skype Translator [4]) to improve technical support in human-machine communication (nowadays the first questions for technical support are being managed by conversational agents). The importance of NLU is also witnessed by the various cloud-based platforms proposed by the major IT companies such as Facebook (i.e., with wit.ai [5]), Google (i.e., with Dialogflow [6]), IBM (i.e., with Watson Conversation [7]) Microsoft (i.e., with LUIS [8]) and so on. Thanks to Cloud Computing, these NLU platforms can be easily accessible from everywhere, they can exploit huge computational power (as provided by the biggest IT companies like Google), and they are ready-to-go, always updated, without any software to install of hardware requirement to satisfy.

This paper compares the most relevant NLU cloud platforms from two points of view: descriptive and performance-based. To the best of our knowledge, this is the first attempt to fully compare and analyze the most important cloud-based NLU tools. For the descriptive analysis, the paper proposes a taxonomy to represent and explore different NLU solutions. The taxonomy is, then, applied to six NLU platforms, for which we discuss their specific characteristics. Furthermore, for the performance-based analysis, we compare the performance of the best three NLU tools analyzed through an experimental evaluation and we discuss their relative strengths and weaknesses.

The remainder of the paper is organized as follows. In Section II, we review existing work, while in Section III, we describe the most relevant Natural Language Understanding cloud platforms. Then, in Section IV, we discuss the taxonomy used to analyze the NLU tools, and in Section V, we evaluate the weaknesses and strengths of such tools. Finally, in Section VI, we conclude the paper.

II. Related Works

In 1972, T. Winograd wrote a paper entitled "Understanding natural language" [9], which describes a computer system for understanding English. The paper describes in details the main components of the system (such as a parser, a recognition grammar, a semantic analyzer, and a general problem solving system) and it proposes a sample parsing program. These components are still the key components of the modern NLU system and only in the last 10 years they have been fully implemented with the advent of a new technology called *conversational assistant* (or "ChatBot"): a computer program designed to interact with users via textual or auditory methods using NLU systems, typically hosted in the cloud. Despite the spread of chatbots in many contexts (for example, customer care support, food order, shopping assistant, reservation) and the increase of NLU systems, to the best of our knowledge the scientific literature has not proposed yet a complete comparison concerning the main cloud NLU system available nowadays. As a matter of the fact, in [10] the authors discuss how it is possible to implement a conversational assistant by using three cloud providers (in particular, Microsoft Azure, IBM Watson, and Heroku) but they do not provide any experimental evaluation for the adopted NLU system. In [11], the authors describe ideas to extend the way of verbal lecture visually and verbally. For the visualization of lecture talk, they created a chatbot by using DialogFlow (formerly called api.ai) without a preliminary comparison with other NLU systems. Furthermore, there are specific blogs, bulletin boards, and other web resources focused on chatbots that provide a very superficial comparison with a few of those NLU systems. In particular, in [12] the authors make a comparison with just two of them (i.e., DialogFlow and wit.ai) by considering a specific task: order a pizza. Finally, in [13], the authors compare 4 NLU platforms only from a performance-based point of view without providing a taxonomy to represent and explore the difference between the NLU solutions. In this work, we plan to fill this gap by providing an exhaustive and complete comparison of the main cloud-based NLU systems available nowadays, according to a proposed taxonomy.

III. Natural Language Understanding Platforms

In the last 10 years, several companies opened their cloud-based NLU tools with the aim of allowing developers and engineers (a) to extend the capabilities of their own NLU products or (b) to create new conversational assistants with ease.

In the former case, four actors dominate the NLU panorama: Apple with Siri, Google with Google Assistant, Amazon with Alexa, and Microsoft with Cortana. These companies release APIs and libraries to extend the capabilities of their respective conversational assistants: as an example, Apple provides *SiriKit* for iOS app developers, while Amazon allows developers to create new *skills* for Alexa. In this paper, we are not interested in investigating and comparing these NLU tools since they are pre-trained for end-users and continuously improved by their respective companies.

We would, instead, compare and critique the latter case, i.e., cloud-based NLU platforms personalized and trained by developers. In this way, we aim at evaluating the "bare" algorithms and cloud services. In this category, we identify six main NLU cloud platforms: 1) Google's *DialogFlow* [6], 2) Facebook's *wit.ai* [5], 3) Microsoft *LUIS* [8], 4) IBM *Watson Conversation* [7], 5) *Amazon Lex* [14], and 6) *Recast.ai* [15].

All these platforms are powered by machine learning algorithms that are totally transparent for the developers. They share some common functionality (e.g., they are cloud-based, they support various programming languages and different natural languages, etc.) but differ significantly in other aspects.

They rely on two concepts for performing NLU operations: **intent** and **entity**. An *intent* represents a mapping between what a user says and what action should be taken by the chatbot. It represents a portion of a conversation. An *entity*, instead, is a tool for extracting parameter values from natural language inputs. Any important data you want to get from a user's request, will have a corresponding entity. To better understand, the sentence "What is the weather in Paris?" is the input part of the intent for *asking the weather*, while "Paris" is an individual of a possible entity named *City* that collect the names of any city in the world. Such NLU platforms allow both the definition of new intents and new entities, as well as the reuse of existing intents and entities (called *pre-build intents* and *pre-build entities*, respectively). In the remainder of this Section, we describe the main characteristics of each NLU platform analyzed.

A. DialogFlow

DialogFlow, previously known as *api.ai*, is a NLU cloud platform owned and maintained by Google. It is a free to use conversational platform, i.e., it is possible to refine a dialog without repeating the context of the conversation. It supports various languages, different programming languages, and it has a series of built-in integration with other chatbot-based platforms (e.g., Telegram, Google Assistant, Amazon Alexa).

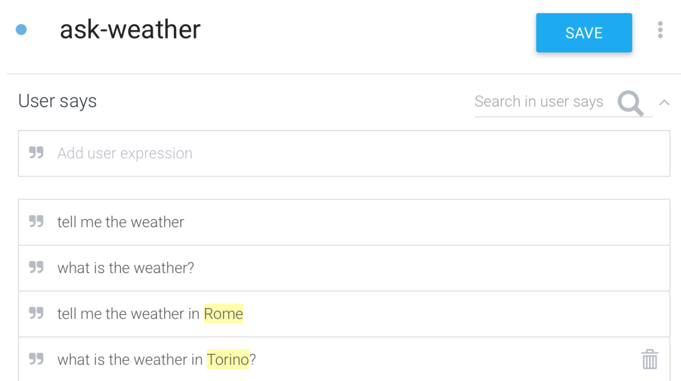

Figure 1. An example of the DialogFlow composition interface: speaking about the weather.

A developer interested in creating a new conversational assistant, or in expanding the conversational capabilities of their existing chatbot, can use one or more forms in a web-based user interface (Figure 1). Each form represents a portion of the conversation, structured as a users' question and some answers. In the DialogFlow form, the developer has to insert some examples of input sentences (i.e., what the portion of the dialog is about) and the "answers" provided by the NLU tool. Answers can be inserted directly into the web interface or can be provided by an ad-hoc server application through the *webhook* mechanism, enabled by the DialogFlow APIs. It is possible to mix, in a single intent, multiple languages.

B. wit.ai

wit.ai is a NLU cloud platform owned and maintained by Facebook. Like DialogFlow, it is a free to use platform, with support of several natural languages but only 3 programming languages.

Differently from all the other NLU tools examined here, wit.ai focuses on extracting meaning from single sentences. In other words, it acts more as a NLU parser than a complete NLU platform: in fact, wit.ai does not provide any integration with other chatbot-based platform, any web interface for handling portions of a conversation, nor any mechanism for maintaining the context through a conversation. All these and other features are in charge of the developers, that should realize in their own code any integration, conversational aspects, etc., in which they are interested.

C. LUIS

Language Understanding Intelligent Service (LUIS) is the NLU platform of Microsoft. Differently from the previous two tools, it is part of the Azure cloud services. Being part of Azure, LUIS shares the pricing schema with it and can access to some additional features. It supports various languages, but only 4 SDKs are available (i.e., C# SDK, Python SDK, Node.js SDK, and Android SDK). All its applications are centered on a domain-specific topic or are content related. Active learning technology is one of LUIS's features. It is possible to use pre-existing, world-class, pre-built models from Bing and Cortana. Models deployment to an HTTP endpoint is a one-click operation; it returns easy-to-use JavaScript Object Notation (JSON) documents. LUIS offers a set of programmatic REST APIs that can be used by developers to automate the application creation process.

D. Watson Conversation

Watson Conversation is the NLU platform of IBM, part of the IBM Bluemix cloud services. It shares Bluemix pricing schema and may access to additional features. It is a conversational platform with the support of various programming and natural languages: it is built on a neural network (one billion Wikipedia words), understands intents, interprets entities and dialogs. It achieves accuracy by attempting to assess as much context as possible. It gets that context both within the passage of the question and from the knowledge base (called a corpus) that is available to it for finding responses. Watson teases apart the question and potential responses in the corpus, and then examines it and the context of the statement in hundreds of ways. Watson then uses the results to gain a degree of confidence in its interpretation of the question and potential answers.

E. Amazon Lex

Amazon Lex is the NLU platform part of the Amazon Web Services (AWS). It shares the pricing schema with AWS and may access to additional features. It is a conversational platform that supports various programming languages but only one natural language (i.e., English). Amazon Lex is a service for building conversational interfaces into any application using voice and text. In particular, it provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text, to enable developers building applications with highly engaging user experiences and lifelike conversational interactions. Amazon Lex exploits the same deep learning technologies that power Amazon Alexa.

F. Recast.ai

Recast.ai is the only analyzed NLU platform owned and maintained by a startup. It is a NLU cloud platform, free to use, with support for multiple natural and programming languages. Recast.ai allows the creation of complex conversational assistants (e.g., like DialogFlow) but also provides a series of dedicated APIs/web services to perform textual and lexical analysis (e.g., like wit.ai).

Figure 2. An example of the block-based interface available from Recast.ai.

To create a conversational assistant, a developer should define an intent in a two-step process: with a form-based interface, they have to define the input sentences (with their entities) of an intent; then, with a block-based interface (Figure 2), they can define the other elements of the intent (chatbot responses and actions) as well as the connections with other intents. It is possible to mix, in a single intent, multiple languages.

Recast.ai is the only platform that allows a developer to publicly share their intents and entities, to be used by other users of the Recast.ai community.

IV. NLU Taxonomy

To describe the *core features* of cloud-based NLU tools, we propose a taxonomy that focuses on 13 different facets. The taxonomy aims at providing a descriptive overview of such NLU platforms, thus enabling a comparison of different tools. In particular, we validate the taxonomy by describing and comparing the six NLU platforms briefly presented in the previous Section. The taxonomy with the description of those six NLU tools is reported in Table I.

The taxonomy focuses on 13 facets, ranging from supported languages to pricing, to advanced and automatic NLU features like fallback intents.

1. **Usability** indicates the perceived usability and ease of use of the user interface provided by the NLU platform. It can assume the following values: a) *high*, simple and intuitive for a developer; b) *medium*; c) *low*, difficult to use for a developer and without supporting documentation.
2. **Languages** reports how many natural languages the platform supports. It also indicates whether those languages can be mixed inside a single intent.
3. **Programming Languages** shows how many programming languages the platform officially supports and maintains.
4. **Pre-build Entities** indicates how many pre-build entities the NLU tool offers.
5. **Pre-build Intents** reports how many pre-build groups of intents the NLU platform offers.
6. **Default Fallback Intent** indicates whether the platform has a fallback mechanism for intents. The *fallback mechanism* allows the proper classification of sentences that are not recognized as part of an existing intent; without a dedicated fallback mechanism, every sentence will belong to a defined intent. For example, suppose to have only one intent that is related to the weather and the sentence "My name is Mark"; without a fallback mechanism the sentence will be classified as pertaining to the "weather" topic.
7. **Automatic Context** reports whether the NLU platform can automatically manage the context in a conversation or it is left to the developer's code.

8. **Composition Mode** shows which is the composition modality adopted by the tool (e.g., form-based, block-based).
9. **Online Integration** reports which third-party integrations are available in the cloud platform.
10. **Webhook/SDK Availability** indicates whether and how a developer can integrate his/her conversational assistant with other software, independently from the available online integrations.
11. **All-in Platform** reports whether the NLU tool is a platform able to provide multiple NLU-related services by its own, i.e., it is not too basic or part of a family of different web services.
12. **Linkable Intents** indicates if it is possible to link one intent to another, directly in the platform.
13. **Price** is the pricing for using the NLU platform.

As detailed in Table I, most of the analyzed NLU platforms show a good usability and support various languages as well as programming languages, so that they can be easily used by developers. They mainly differ, instead, in the *automatic* features that pertain to NLU: default fallback intent, automatic context handling, and linkable intents. These three aspects are fundamental for allowing a developer to get started immediately in creating their own conversational assistant. From a descriptive point of view, we can assure that DialogFlow is the most complete NLU platform since it provides the best solutions for the majority of facets considered in the taxonomy.

V. Experimental evaluation

To conduct a performance evaluation of the NLU platforms, we focused on those tools that exhibit an automatic handling of NLU-related features. In particular, we selected the three NLU platforms with a default fallback intent, so that the handling of unknown sentences is delegated to the platform's algorithm and not to the developer's experience and code.

We evaluated DialogFlow, LUIS, and Watson Conversation concerning two intents: a *weather* intent, created equally in the three platforms, and the *default fallback* intent already provided by each NLU tool. The goal of the evaluation was to understand at which extent each platform detects the intention of the user and the parameters in the sentences. In other words, we were interested in whether the NLU platform placed a sentence in the "right" intent, and whether all the entities present in the trained intent were identified. The language selected for the evaluation was English.

A. Procedure

We created a "weather" intent and enabled the *default fallback* intent in each of the three NLU platforms. Then, for each NLU tool, we trained the intent with the following five sentences (five is the minimum number of training sentences suggested by NLU platforms documentation, such as Watson):

1) Should I take an umbrella outside today?
2) Show me the forecast for next Monday
3) What's it like outside my office?
4) Is it sunny in Rome today?
5) What will be the weather tomorrow in San Francisco?

For each sentence, we highlight to the NLU tool the presence of specific entities and whether they are mandatory or not. In particular, we used two pre-build entities for *date* and *location*, and we instructed the platforms to require the location for every sentence of the "weather" intent. For the *date* entity, we selected the "@sys.date" entity in DialogFlow, "datetimeV2" in LUIS, and "@sys-date" in Watson. For the *location* entity, instead, we selected the "@sys.location" entity in DialogFlow, "geography" in LUIS, and "@sys-location" in Watson (marked as a beta entity). If the *location* entity was absent from a sentence, we trained each NLU platform to ask "Where?" before providing a fixed answer (i.e., "It will be sunny in {location}").

To exemplify the workflow, if a user submit the sentence "*Should I take an umbrella outside today?*", the NLU platform should recognize that the sentence belongs to the "weather" intent and ask "*Where?*", since no location is present in the original sentence. The user, at this point, should indicate a location (e.g., "*In Rome*") and the platform will reply "*It will be sunny in Rome*."

After this training procedure, we submitted 24 sentences to each NLU platform, by paying attention to reset the context in those platform with an automatic context management. By doing so, we assure that each sentence is evaluated independently from the previous one. The 24 sentences were divided in the following way: (a) 3 sentences were taken from the training set, (b) 13 sentences were about the weather, (c) 8 sentences were not about the weather. For the last set of sentences, we chose various non-weather sentences considering three categories: (i) sentences that are completely not related to the weather (i.e., "Ice cream"), (ii) questions with cities names (i.e., "What is the time in Venice now?") and, finally, (iii) general questions/greetings (i.e., "Which information may you provide?"). The answer provided by the platform in these cases can be decided by the developers (for example, the answer could be "Could you reform your request, please?".).

We stored the results provided by each platform in terms of response, accuracy of the recognition, and classified intent. The full list of sentences, split in the three areas, along with the main results about intent classification are reported in Table II. In the Table, we highlighted in bold the wrong results (intent name and accuracy).

B. Results and Discussion

As mentioned before, we split the full list of sentence in 3 areas: (i) sentences from the training set, (ii) sentences about weather, and (iii) sentences not related to weather. In this section we discuss the results obtained by the NLU platforms for each area.

For the first area, as expected, all NLU platforms were able to detect the right intent (i.e., weather) with maximum confidence level (1.0). Conversely, for the second area, we noted significant differences between the NLU platforms behavior. In particular, DialogFlow detected the default intent for 8 sentences (out of 13) instead of the weather one. The other two NLU platforms performed better since they identified the weather intent with an high confidence level for all the 13 sentences. In particular, LUIS returned an high accuracy (i.e., $\geq$ 0.99) for all sentences while Watson returned a slightly

TABLE I. THE DESCRIPTIVE TAXONOMY FOR THE SIX NLU PLATFORM EXAMINED

Platform	*Usability*	*Languages*	*Programming Languages*	*Pre-build Entities*	*Pre-build Intents*	*Default Fallback Intent*	*Automatic Context*	*Composition Mode*	*Online Integration*	*Webhook/ SDK Availability*	*All-in Platform*	*Linkable Intents*	*Price*
Dialog-Flow	High	15, from English to Chinese	11, from Java to Ruby	60, from addresses to colors	34, from small talks to currency converters	Yes	Yes	Form-based	14, from Telegram to Alexa	Webhook and SDKs	Yes	Yes	Free
wit.ai	Medium	50, from Albanian to Ukrainian	3: Node.js, Python, and Ruby	22, from location to email	Zero	No	No	Form-based	Zero	SDK	No	No	Free, contact heavy usage
LUIS	Medium	10, from English to Chinese	4: Android, Python, Node.js, and C#	13, from numbers to geography	20, from calendar to fitness	Yes	No	Form-based	Zero	Webhook and SDK	No, other services are in Azure	No, other services are in Azure	Free up to 10k requests per month
Watson Conversation	High	12, from English to Chinese	6, from Node.js to Java	7, from time to person	Zero	Yes	Yes	Form and block-based	Zero	SDK	No, other services are in Bluemix	Yes	Free up to 10k requests per month
Amazon Lex	Low	1: English	9, from Java to Go	93, from Alexa	15, from Alexa	No	Yes	Form-based	3: Twilio SMS, FB Messenger, and Slack	SDK	No, other services are in AWS	Yes	Free for the 1st year (with limits)
Recast.ai	Medium	16 at the standard level	7, from Python to Go	31, from colors to distance	3, several from the community	No	No	Form and block-based	Zero	Webhook and SDK	Yes	Yes	Free

TABLE II. THE MAIN RESULTS OF THE EVALUATION: THE RECOGNIZED INTENTS WITH THEIR ACCURACY

Sentences	*DialogFlow*	*LUIS*	*Watson*
What will be the weather tomorrow in San Francisco?	weather (1.0)	weather (1.0)	weather (1.0)
Show me the forecast for next Monday	weather (1.0)	weather (1.0)	weather (1.0)
What's it like outside my office?	weather (1.0)	weather (1.0)	weather (1.0)
Is it cold outside?	**default**	weather (0.99)	weather (0.85)
What will the weather be like in Paris?	weather (0.71)	weather (0.99)	weather (0.91)
What's the weather like in 3 days?	weather (1.0)	weather (0.99)	weather (0.92)
Is it windy today?	**default**	weather (0.99)	weather (0.88)
I want the weather in New York, in two weeks	weather (0.39)	weather (0.99)	weather (0.26)
Is it raining today?	**default**	weather (0.99)	weather (0.88)
What will the weather be like at my vacation home this weekend?	**default**	weather (0.99)	weather (0.92)
What is the weather like near my next meeting?	**default**	weather (0.99)	weather (0.92)
Is it a beautiful day for a walk?	**default**	weather (0.99)	weather (0.85)
Is the weather good for a walk today?	**default**	weather (0.99)	weather (0.90)
Should I take an umbrella in Oslo, tomorrow?	weather (0.90)	weather (1.0)	weather (0.96)
What is the weather for the next three days in Dublin?	weather (0.54)	weather (0.99)	weather (0.93)
Is it raining in San Diego now?	**default**	weather (0.99)	weather (0.83)
What is the time in Venice now?	default	**weather (0.99)**	**weather (0.37)**
What is the timezone of New York?	default	**weather (0.99)**	**weather (0.37)**
Hello!	default	**weather (0.99)**	default
My name is Mark.	default	**weather (0.99)**	**weather (0.25)**
Ice cream	default	**weather (0.99)**	default
<a word with random chars>	default	default	default
I need help	default	**weather (0.99)**	default
Which information may you provide?	default	**weather (0.99)**	default

lower accuracy (i.e., ≥ 0.85) for the majority of the sentences. It is worth noticing that the sentence with the lowest accuracy is "*I want the weather in New York, in two weeks*" for both DialogFlow (with accuracy = 0.39) and Watson (with accuracy = 0.26). In this second area, LUIS was the NLU platform which performed better than the other two. In the last set of sentences, the performance of the tools varied considerably. In particular, DialogFlow was the only tool which correctly detected the default intent for all sentences. As a matter of fact, both LUIS and Watson detected the weather intent for some sentences. In particular, LUIS performed badly: only the word with random chars was detected as default intent, in all other case, LUIS detected the weather intent with high accuracy (i.e., ≥ 0.99). Watson performed slightly better than LUIS: for three sentences it detected the weather intent, albeit with a low accuracy (i.e., ≤ 0.37). As mentioned before, concerning the entity identification, all NLU platforms were able to recognize them with 1.0 level of accuracy.

In general, we can say that Watson is the best NLU platform in our study since it detects wrong intents only for 3 sentences over 24. Moreover, for these 3 sentences it provides a low accuracy level that can be exploited by a developer to improve the Watson's intent detection algorithm.

VI. CONCLUSION AND FUTURE WORKS

The idea of being able to hold a conversation with a computer has fascinated people for a long time and has featured in many science fiction books and movies. With recent advances in spoken language technology, artificial intelligence, and conversational interface design, coupled with the emergence of smart devices, it is now possible to "ask" these devices to perform many tasks by using natural language [16]. In this paper, we compared and critiqued the main cloud platforms for natural language understanding from a descriptive and performance-based point of view. In particular, we proposed a taxonomy that focuses on 13 different characteristics of a NLU platform and we then conducted a functional evaluation to understand at which extent each platform is able to detect the intention of the user. From the results, we can affirm that Watson is the platform who performs best since it is able to assign the right intent in the majority of the cases studied and, moreover, even when it detects the wrong intent, the accuracy level is low so that a developer can exploit this information to train Watson to improve its intent detection algorithm. As future work, we plan to extend our experimental evaluation by considering many different intents at the same time and we will evaluate how the NLU platforms exploit the context in a more complex conversational dialog.

REFERENCES

[1] M. Galeso, Apple Siri for Mac: An Easy Guide to the Best Features. USA: CreateSpace Independent Publishing Platform, 2017.

[2] A. Nijholt, "Google Home: Experience, support and re-experience of social home activities," Information Sciences, vol. 178, no. 3, 2008, pp. 612–630.

[3] J. Hirschberg and C. D. Manning, "Advances in natural language processing," Science, vol. 349, no. 6245, 2015, pp. 261–266.

[4] W. D. Lewis, "Skype translator: Breaking down language and hearing barriers," Translating and the Computer (TC37), vol. 10, 2015, pp. 125–149.

[5] "The wit.ai project," https://wit.ai, accessed: 2017-11-01.

[6] "The Dialogflow project," https://dialogflow.com, accessed: 2017-11-01.

[7] R. High, "The era of cognitive systems: An inside look at IBM Watson and how it works," IBM Corporation, Redbooks, 2012.

[8] "The LUIS project," https://luis.ai, accessed: 2017-11-01.

[9] T. Winograd, Understanding Natural Language. Orlando, FL, USA: Academic Press, Inc., 1972.

[10] P. Amit, K. Marimuthu, R. A. Nagaraja, and R. Niranchana, "Comparative study of cloud platforms to develop a chatbot," International Journal of Engineering & Technology, vol. 6, no. 3, 2017, pp. 57–61.

[11] S. Matsuura and R. Ishimura, "Chatbot and dialogue demonstration with a humanoid robot in the lecture class." Lecture Notes in Computer Science, vol. 10279, 2017, pp. 233–246.

[12] "Api.ai vs wit.ai (or is it Google vs Facebook?)," https://www.themarketingtechnologist.co/api-ai-vs-wit-ai/, accessed: 2017-11-01.

[13] D. B. et al., "Evaluating natural language understanding services for conversational question answering systems," in Proc. of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017, pp. 174–185.

[14] "Amazon Lex," https://aws.amazon.com/lex/, accessed: 2017-11-01.

[15] "Recast.ai," https://recast.ai, accessed: 2017-11-01.

[16] M. McTear, Z. Callejas, and D. Griol, "The conversational interface," New York: Springer, vol. 10, 2016, pp. 978–3.