Protecting Privacy and Open Competition with Almond: An open-source virtual assistant

Will Alexa and Google Assistant become the duopoly platforms on which consumers reach web services and IoTs verbally? With open and collaborative research, we can build the best open-source virtual assistant to ensure choice, privacy, and open competition.

By Monica S. Lam, Giovanni Campagna, Silei Xu, Michael Fischer, and Mehrad Moradshahi
DOI: 10.1145/3355757

With the proliferation of the smart internet of things (IoT) and millions of web services, many of our personal or professional tasks can be accomplished digitally today. While in the past humans would learn how to interface with the computer, computers are now learning the human language. Today, commercial virtual assistants such as Amazon Alexa, Apple Siri, and Google Assistant, are widely adopted. Alexa today can understand natural language commands for more than 90,000 skills, which are linguistic user interfaces (LUI) to third-party web services or IoT devices [1]. The assistant will mature from simple commands to knowing about us intimately and performing complex tasks.

We have developed an open-source virtual assistant, called Almond, that lets users compose commands that connect different resources [2]. Consider, for instance, a day in the life of an asthma patient, who we will call Bob. He can say to his Almond assistant, “Log the locations where I use my inhaler” to record the places that trigger asthmatic reactions, or “Tell my Dad whenever I am taken to the hospital.” Similarly, his doctor can tell Bob’s Almond assistant that “Whenever Bob’s peak flow meter drops below 180L/min, let me know,” and “If Bob is running where the air quality index is above 100, tell him to stop.” As open-source software, Almond can run on Bob’s own devices and discloses information only to whomever Bob specifies.

This example illustrates why we need to enable each person to use plain language to express their unique needs, which may be personal or professional. Doctors can provide prompt health care cost-effectively by focusing on patients in need. The continuous monitoring of outcomes and their correlation with treatment across patients will provide invaluable data to improve...
health care and make automated quality health care available, especially to the underserved population.

The virtual assistant will transform our digital experience by giving us a fully personalized and integrated linguistic interface for our digital assets, which are currently siloed in different services. Furthermore, as it collects and analyzes detailed information across all users, it will learn an accurate model of human behavior. It will predict and intervene with our behavior such as reminding us to take our medication. Similarly, by gathering details on business decisions and outcomes, assistants will monitor and optimize business logic across professions.

THE POTENTIAL THREATS OF VIRTUAL ASSISTANTS

As we develop technology for virtual assistants, we should also be aware of potential downsides. It is likely a platform monopoly or duopoly will emerge for the virtual assistant. Monopolies hurt consumers as they stifle competition and innovation. A virtual assistant monopoly is particularly worrisome because it controls consumers’ access to the digital world and sees the private data of billions of people across all different services.

A proprietary linguistic web. We use the virtual assistant to access the linguistic web, just like how we use the browser to access the graphical web. However, while the graphical web is non-proprietary, we are witnessing the creation of proprietary linguistic webs. It is hard to understand one natural language, let alone the many languages needed for the international market. The state-of-the-art neural network approach requires a large volume of annotated human sentences; hence, Amazon has 10,000 employees devoted to Alexa [3]. Moreover, popular assistants attract vendors, which, in turn, attract more users. Thus, a significant barrier to entry, due to the network effect and high development cost, will likely lead to a virtual assistant monopoly or duopoly.

By intermediating between consumers and the web, virtual assistants have the power to channel users to their own or promoted products. They may even charge a fee to commercial
transactions conducted on their platform, similar to app stores charging mobile app developers 30 percent of their revenues. More importantly, by getting access to users’ accounts, virtual assistants are privy to valuable business intelligence data. For example, users’ thermostat accounts contain valuable energy usage data. A virtual assistant monopoly will have an unfair advantage in all consumer businesses.

**Monopoly platforms threaten privacy.** There is a growing awareness in both the U.S. and the E.U. that large platforms pose severe threats to individual privacy. The provision of the General Data Protection Regulation (GDPR) by the E.U. represents the first systematic attempt to address the existing dominance large platforms have over personal information. For example, the social networking market is dominated globally by Facebook, which has no meaningful competition today in most countries. Companies owning billions of people’s personal information have tremendous power. They can share users’ data with other parties, influence users’ opinion in important decisions such as presidential elections and intervene with users’ behavior. A monopoly assistant platform will have access to data in all our different accounts; they will have more knowledge than Amazon, Facebook, and Google combined.

**AN OPEN VIRTUAL ASSISTANT MANIFESTO**

It is essential that we lower the barrier to entry to virtual assistants, increase innovation, open competition, and give consumers a choice. We can summarize our vision with the following manifesto:

1. **Democratize AI for linguistic user interfaces.** We should have open, collaborative research to put the best linguistic technology in the hands of all businesses.

2. **An open non-proprietary linguistic web.** All skills, or linguistic user interfaces, should be made available to any virtual assistant.

3. **Sharing with individual data ownership.** Consumers should have a choice in virtual assistant services and the ability to control how we share our data.

**ALMOND: AN OPEN-SOURCE VIRTUAL ASSISTANT**

The Almond virtual assistant is an open-source framework designed to fulfill the vision in our manifesto. However open source alone does not attract users, functionality does.

**Programming the assistant in natural language.** Commercial assistants translate natural language into an intermediate representation that matches the semantics of the sentences closely, and then infer the proper action separately. For example, the Alexa Meaning Representation Language is associated with a closed ontology of 20 domains, each manually tuned for accuracy. Semantically equivalent sentences have different representations, requiring complicated and expensive manual annotation by experts, who must know the details of the formalism and associated ontology [4]. The ontology also limits the scope of the available commands, as every parameter must be an entity in the ontology (a person, a location, etc.) and cannot be free-form text. This approach is expensive and hard to scale.

The Almond design is based on the concept of natural language programming: Almond accepts natural language commands and translates them to executable programs. While there are numerous natural languages, all with ill-defined semantics, a formal programming language can succinctly and fully capture the capability of a virtual assistant. We have created a new language specifically for this purpose, called “ThingTalk,” which consists of simple constructs that invoke skills in a repository called “Thingpedia.”

The virtual assistant will transform our digital experience by giving us a fully personalized and integrated linguistic interface for our digital assets.
have also developed a deep-learning neural network, LUInet (linguistic user interface network), to translate natural language into ThingTalk code, which is executed by the assistant.

This approach has several advantages. First, it eliminates the need and inefficiency of an intermediate representation that is sensitive to the choice of input natural languages. Furthermore, the ThingTalk code can also be converted back into a canonical natural language sentence to confirm the code before it is run.

Second, programming languages are compositional. A relatively simple construct can connect different skills from Thingpedia to provide many useful functions. For example, all the different commands that Bob issues in the previous asthma example can be expressed by the same when-get-do construct; when an event occurs, get some data, and do an activity. Each of these clauses can be qualified, e.g., the air quality index must be above 100 before alerting Bob. These commands are a superset of what is supported by the if-this-then-that (IFTTT) service. Millions of combinations can be supported by a skill repository with hundreds of functions.

Note that to support interoperability and compositionality, Thingpedia captures the full signatures of APIs, which include both input and output parameters. Commercial assistants, on the other hand, are intent-based and capture only the input parameters. Thus, they cannot support natural language programming.

Third, this approach makes it easy to extend the functionality of the assistant by reducing the cost of manual annotation of natural language sentences. We have developed a tool called Genie [5] that can generate labeled training data (natural language sentences and corresponding formal programs) for new constructs and new skills. Developers can write templates to create a large variety of synthesized data, only a sample of which is shown to crowdsourcers to paraphrase. Genie lets developers define skills and constructs suitable for their domain and get a natural language interface without machine-learning expertise. In our experiment with about 130 functions in Thingpedia, our neural model yields an accuracy of about 68 percent on realistic when-get-do commands consisting of up to two functions and two filters.

Access control. Many services today demand ownership of users’ data and offer limited ways in which users can share their data. Almond provides users with fine-grain control over who, what, when, where, and how their data are to be shared [6]. For example, a patient can say, “Share my CT scans with academic research institutes, provided my personal identity information is removed.” A daughter can say, “Let my dad see my security camera if there is motion, only if I am not home.” A parent can say, “Let my son watch PG-13 movies on Netflix between 7-9pm.” A university administrator can say, “Send all Facebook posts on the university page to students without Facebook accounts.”

Almond translates access control expressed in natural language into formal ThingTalk code. The owner’s assistant executes these commands and only shares the results with the specified individuals. This approach lets us share anything that Almond can perform with anybody we wish. We found through a user study that twice as many people are willing to share their assets if they can impose constraints like the ones above. ThingTalk is expressive enough to support 85 percent of the constraints suggested by 60 users in a survey.

Access controls are represented as SMT (satisfiability modulo theories) formulas in our language implementation. We use verifiably correct SMT algorithms to ensure conformance of constraints and to synthesize conforming commands in case the original request is not allowed. For example, the teen can add the constraint “only if I am not home” to the request to see the security camera.

To facilitate interoperability between virtual assistants, we propose a remote ThingTalk execution protocol. A user’s assistant can route a data request to the owner’s assistant, which will execute the command upon approval of the owner and return the result.

This architecture makes possible a federation of virtual assistants, eliminating the need for a centralized service that sees all data.

**Making the Manifesto Come True**

Open-source technology has shown to be effective in offering an alternative to monopolies. Unix, first released in 1971, and subsequently BSD and Linux offered an alternative to Windows and led to the server and mobile operating systems used widely today. The NCSA Mosaic browser, first released in 1993, and subsequently Netscape and Firefox, offered a widely adopted open-source alternative to Internet Explorer. The success of AlexNet, VGG, Inception, ResNet for object recognition demonstrates the importance of open research in machine learning. More recently, the Stanford OpenFlow project gave rise to the highly successful Software Defined Networking (SDN) movement. By creating an open programmable standard in communications, SDN opened up the competition, allowing many companies to innovate and compete, and thus invigorated a previously ossified networking industry dominated by Cisco.

We believe Almond can attract collaborators as it protects consumer privacy and lets companies own their interfaces, customer relationships, and corporate intelligence. Here we describe how we can build upon the Almond research prototype to make the open-assistant manifesto come true.

**LUInet: Democratizing AI for Linguistic User Interfaces.** In about five years, we predict there will be a LUInet that is proficient in understanding verbal instruction of all digital tasks. The only question is whether it is propri-
Almond can attract collaborators as it protects consumer privacy and lets companies own their interfaces, customer relationships, and corporate intelligence.

Virtual assistants will revolutionize how we interact with computers. The data collected by assistants can be used to predict and intervene with our behavior and to automate professional processes. The massive barrier to entry for virtual assistants means that a monopoly or duopoly platform is likely to emerge unless we work together to create an open-source alternative.

We have developed Almond, a collaborative virtual assistant framework that can integrate linguistic interface capabilities developed independently by domain experts. We have also developed a distributed architecture that lets users use natural language to control who, what, when, where, and how their data are to be shared.

By empowering companies and individuals, we hope to create (1) the best and open-source deep neural network for linguistic user interfaces (LUInet), (2) an open, non-proprietary, interoperable linguistic web (Thingpedia), and (3) an open-source federated virtual assistant that lets users control data sharing without a centralized service (Almond). Together we can create an ecosystem where businesses compete openly, users can maintain their privacy, and data is collected with user approval for advancing big-data science.

We invite you to get involved with Almond by visiting https://almond.stanford.edu/ and contributing to our GitHub project: https://github.com/stanford-oval/StanfordOpenVirtualAssistantLab.

Acknowledgement

Funded in part by the National Science Foundation under Grant No. 1900638.

References

[1] Perez, S. Alexa skills top 80,000 after a big Alexa-powered holiday season. Tech Crunch (February 2019); https://techcrunch.com/2019/02/01/alexa-skills-top-80000-after-a-big-alexa-powered-holiday-season.


Biographies

Monica S. Lam is a professor in the Computer Science Department at Stanford University since 1988. She received a B.Sc. from the University of British Columbia in 1980 and a Ph.D. in computer science from Carnegie Mellon University in 1987. Prof. Lam is a Member of the National Academy of Engineering and an Association of Computing Machinery (ACM) Fellow. She is a co-author of Compilers: Principles, Techniques, and Tools (2nd Edition), also known as the “Dragon book.” She is the PI of the NSF Research Award “Autonomy and privacy with open federated virtual assistants.”

Giovanni Campagna is a third year Ph.D. student at the Stanford University Computer Science Department, advised by Prof. Monica Lam. His interests lay at the intersection of programming languages and natural language processing. He is the lead developer of the Almond project.

Silei Xu is a fourth year Ph.D. student in the Stanford Computer Science Department, under the supervision of Prof. Monica Lam. He received his bachelor’s degree from the University of Science and Technology of China and master’s degree from the Chinese University of Hong Kong. His research focuses on making the web, IoTs, and knowledge accessible and programmable with natural language. He is one of the main developers of Almond project.

Michael Fischer is a Ph.D. student at Stanford University in computer science studying artificial intelligence, advised by Prof. Monica Lam. He is interested in programming and making it easier through the use of deep learning. He sees the democratization of technical skills and programming as an essential step in achieving social justice in education and job skills. Before being a Ph.D. student, he graduated from Stanford University with honors in computer science and a minor in mathematics.

Mehrdad Moradshahi is a Ph.D. student in the Stanford Computer Science department advised by Prof. Monica Lam. He has received his bachelor’s degree from Sharif University of Technology with highest honors and master’s degree from Stanford University, both in electrical engineering, in 2016 and 2018 respectively. He has been working mainly on the AI and natural language understanding side of the Almond project since 2016.

© 2019 Copyright held by author. 1528-4972/19/09415.00