Kadia

Release 0.0.1

Jul 08, 2020

Contents

1 Kadia‘s mission **3**
1.1 The concepts . . . 3
1.1.1 Context-based skill activation system . . . 3
1.1.1.1 Example . . . 4
1.1.2 Action sequence recognition . . . 4
1.1.2.1 Example . . . 4
1.1.3 Text style embedding . . . 4
1.1.4 Advanced NER . . . 4
1.1.5 Embedded Skills . . . 4
1.2 Potential design issues . . . 4
1.2.1 Skills names and contexts conflict . . . 5
1.2.1.1 Problem . . . 5
1.2.1.2 Solution . . . 5
1.2.2 RCE . . . 5
1.2.2.1 Problem . . . 5
1.2.2.2 Solution . . . 5
1.3 The architecture . . . 5
1.3.1 User Configs . . . 5
1.3.1.1 User session configs . . . 5
1.3.1.2 User configs . . . 5
1.3.2 Meta Manager . . . 5
1.3.3 Preprocessor . . . 6
1.3.4 Skill Manager . . . 6
1.3.5 Skill . . . 6
1.3.6 Skill Finder . . . 6
1.3.7 Intent Embedder . . . 6
1.3.8 Algo Recognizer . . . 6
1.3.9 Style Enchancer . . . 6
1.4 The path . . . 6
1.4.1 Version 0.1 . . . 6
1.4.2 Version 0.2 . . . 7
1.4.3 Version 1.0 . . . 7
1.4.4 Version 1.0+ . . . 7
1.5 Tech Stack . . . 7
1.6 Do you need help? . . . 8

2 Indices and tables 9

Kadia is open-source intelligent virtual assistant integrated with ASR system.

CHAPTER 1

Kadia‘s mission

The main purpose of the project is to test various concepts and technologies to enhance the quality of the modern IVA industry. We aim to reach it by implementing our own open-source Kadia using publicity-available state-of-the-art technologies. Along with the code itself, we also want to develop standards for components of the assistant such as skills, algorithm pattern recognition, etc..

1.1 The concepts

Kadia in its core is NLP and skill management system integrated with the outer-source ASR module.

Kadia is several dockerized micro services. Thus, it will be modular and more flexible. As an additional bonus, it will make deployment easier.

The project Kadia wants to test these features:

- *Context-based skill activation system*
- *Action sequence recognition*
- *Text style embedding*
- *Advanced NER*
- *Embedded skills*

Along with common:

- The extensible skill system
- Modular easily-configurable design

1.1.1 Context-based skill activation system

Most of today IVA does this: “$IVA, activate $skill_name”. It’s not a natural way of communication for users. In Kadia it’ll be like this: “Kadia, $command”, where command is the description of what to do.

1.1.1.1 Example

if there is a skill for launching nukes at the USA and you for some reason want to do it, you can just say "bomb out the USA".There is no need for script name memorization. Then, if it's important to action (such as starting a nuclear war), Kadia will say the description of the skill, which it's going to activate, and will ask to confirm. Example: "You are going to launch nuclear missiles at the US via the "Red button protocol". Are you sure?"

1.1.2 Action sequence recognition

We want to give the user an option to naturally compose script skills by saying the sequence of skills to run.

1.1.2.1 Example

There are "download media" and "shut down this computer" scripts. You want to download all GoT series at night when any noise (including the noise of coolers) is annoying.

So, you say: "Kadia, download GoT series and shutdown". Kadia will ask you to confirm: "I am going to download the GoT series with a total size of 20Gb. Then, I'll shut down this PC. Ok?"

1.1.3 Text style embedding

The modern IVA lacks the style. As a consequence, they are less adorable by users. We want to change it by enhancing skills texts with one of several styles available for users. Of course, any mistakes and this feature itself are not affordable for important commands. Thus, it'll be optional for both skill and user.

1.1.4 Advanced NER

IVAs with skill support like Alisa from Yandex often support only a few types of named-entities.

1.1.5 Embedded Skills

There are 2 types of skills: scripts and conversations. Scripts do not interact with the user. Conversation may be interrupted for script inference. For example, during conversation about booking a table in restaurant the user can ask Kadia about the weather.

It's our unique feature.

1.2 Potential design issues

Potential design issues and ways to solve them.

- *Skills names and contexts conflict*
- *Potential RCE in cloud and on user device*

1.2.1 Skills names and contexts conflict

1.2.1.1 Problem

If there are skills with similar contexts, the system may be confused.

1.2.1.2 Solution

Sort skills by some score. The score must be calculated by using global run counts and this user run counts along with the user history

1.2.2 RCE

1.2.2.1 Problem

On-device skill execution is dangerous RCE. Cloud RCE must be also taken into consideration

1.2.2.2 Solution

On-device skills are reviewed. Cloud skills do not get access to ruin something.

1.3 The architecture

1.3.1 User Configs

1.3.1.1 User session configs

local If this session is hosted on-device or web-based. This is important for skills, that do something with device.

visual If Kadia is able to show some visuals.

1.3.1.2 User configs

style Which style will be applied to text. None is default option.

confirmation_threshold The threshold of confidence score to ask user to confirm actions.

1.3.2 Meta Manager

Meta Manager is the module responsible for:

- settings
- workflow management

1.3.3 Preprocessor

It parses raw text to unified format used in Kadia‘s ecosystem. It’ll be like Alisa‘s but with references to original text and original text itself. Maybe, it will include correction suggestions.

See https://github.com/flairNLP/flair for NER information.

1.3.4 Skill Manager

This is the module for interaction with a skill. Skill Manager will download the code of skill, run it and capture its variables. All variables will be saved in MongoDB for future runs.

1.3.5 Skill

Skill is the exterior module of Kadia. There are 2 types of skills: conversations and scripts. Scripts can be embedded in conversations, but can not interact with the user. Conversations are skills that consecutively interacts with the user.

Skill is running only when processing the request.

Skill response may be one of these: - text (not an option for scripts) - i do not understand you flag - text with the end of dialog flag

1.3.6 Skill Finder

This is the module for finding skills relevant to the given context. It needs to use an efficient algo, however, at the beginning Sphere-KNN is ok.

1.3.7 Intent Embedder

Probably, it will mask named entities before feeding into something like DistilBERT.

1.3.8 Algo Recognizer

This is the module for algorithms recognition. It outputs actions trees. Algo Recognizer utilizes Skill Finder along with user’s confirmations to parse phrases and determine which part belongs to which skill.

1.3.9 Style Enchancer

This is the module for text stylization. All named entities must be inserted and verified after processing.

1.4 The path

1.4.1 Version 0.1

This version will test our basic systems. Skills will be called by /run commands by the user directly. The main goal is to test skill API, deployment and teamwork.

Available modules:

- Meta Manager
- Parser&NER
- Skill Manager
- 1 conversation and 1 script

Demo:

- telegram bot

1.4.2 Version 0.2

This version will test our context system. Skills will be run by Kadia.

New:

- Skill Finder
- Intent Embedder
- 3 additional conversations and 3 additional scripts

1.4.3 Version 1.0

This is the first public version.

New:

- Algo Recognizer
- Frontend
- Style Enchancer
- 3 additional conversations and 3 additional scripts

Demo:

- telegram bot
- kadia.github.io

1.4.4 Version 1.0+

Feedback based features. Marketing compaign.

New:

- more skills
- fixes.

1.5 Tech Stack

Atlas MongoDB, Docker, Cloud Run (Cloud Functions where it is possible), FastAPI, Python3.6+, PyTorch

1.6 Do you need help?

For any question, please don't hesitate to contact Andrei Ishutin via hazmozavr@gmail.com

CHAPTER 2

Indices and tables

- genindex
- modindex
- search