

Alexa Skills

“Wismarer Nachrichten”

Inhalt

- Idee
- Alexa Basics
 - Invocations, Utterances und Slots
 - Intents
- Umsetzung
 - Dialog Flow
 - Funktionen
 - Klassen
- Erweiterung
- Deployment
- Demo

Idee

- Alexa Nachrichten Skill sind zu eingeschränkt
- Die Nachrichten verderben mir die Stimmung

Alexa Basics: Invocations, Utterances und Slots

- Invocation
 - Schlüsselwort unseres Skills
 - sollte möglichst einmalig sein
 - alles nach Schlüsselwort wird als Utterance gewertet

Wismarer Nachrichten, wie sind die Nachrichten auf Golem?

Invocation

Utterance

Alexa Basics: Invocations, Utterances und Slots

- Utterances
 - Lösen einen Intent aus
 - Können Variablen enthalten
 - Keyword matching basiert
- Slot
 - Variable mit Datentyp

Erzähle mir mehr über das Sentiment in eins.

verworfen

utterance

slot value

```
{
  "name": "Senti",
  "slots": [
    {
      "name": "number",
      "type": "AMAZON.NUMBER"
    }
  ],
  "samples": [
    "Gefühl in {number}",
    "sentiment in {number}",
    "über {number}"
  ]
}
```

Alexa Basics: Intents

- Intents
 - Funktionen
 - werden von Utterances ausgelöst
 - können Slot Values übergeben bekommen

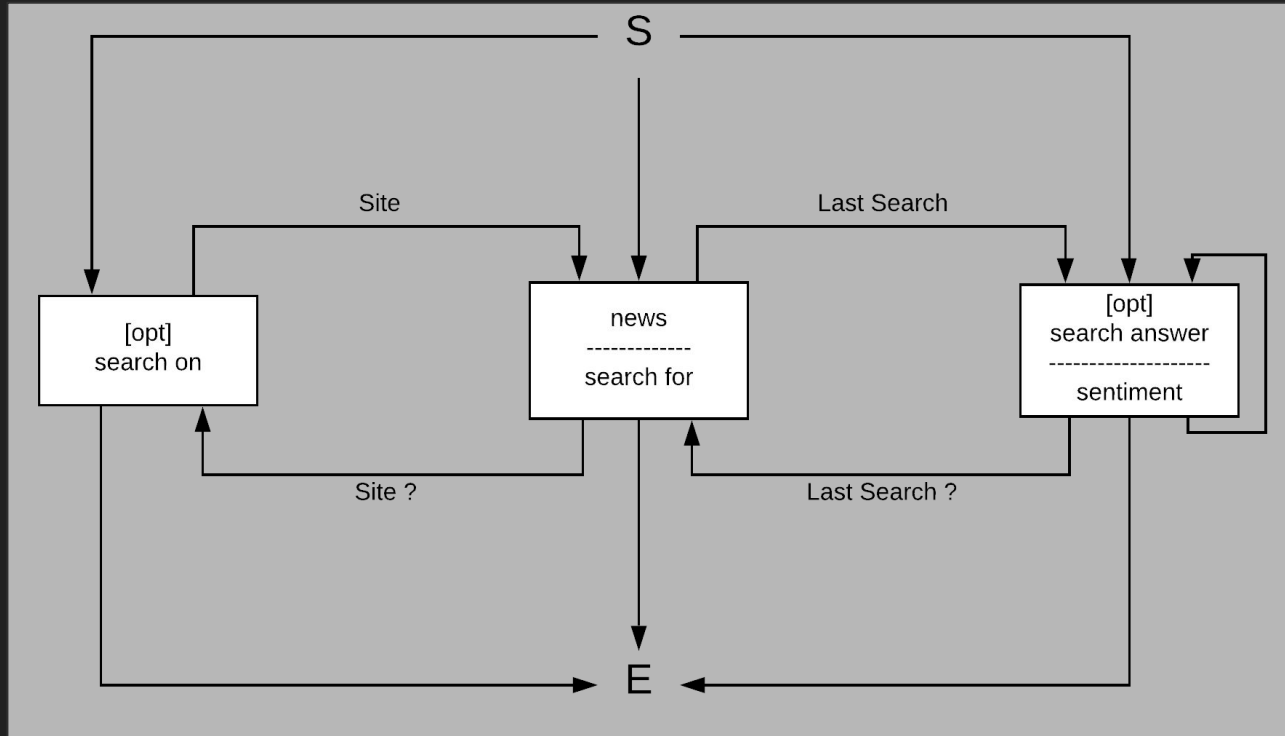
Alexa Ablauf



Umsetzung: Vorüberlegung

- AWS Lambda / Self-Hosted Endpoint
- Amazons Framework / Ask-Flask

Umsetzung: Dialog Flow



Umsetzung: Funktionen

search_on

Setzt die Seite in der Session auf der gearbeitet werden soll.

news

Sucht nach Links zu Artikeln auf der Nachrichtenseite

search_for

Sucht nach Links zu Artikeln in der Suche

search_answer

Liest den ausgewählten Beitrag vor. Der Link zum Artikel wird der Session entnommen.

sentiment

Bewertet das Sentiment des gewählten Beitrags. Der Link zum Artikel wird der Session entnommen.

Umsetzung: Klassen

Site

```
siteName= ""  
baseUrl= ""  
searchURLString = ""  
xpath = dict()
```

```
search_article(topic)  
get_news()  
read_headlines()  
[...]
```

Golem(Site)

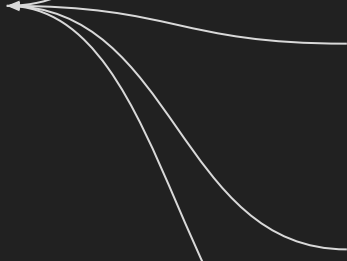
```
siteName = "golem"  
baseUrl = "https://www.golem.de/"  
searchURLString = "https://suche.golem.de/search.php?l=10&q="
```



```
xpath = dict()  
xpath["searchArticle"] = '//span[@class="dh2 head2"]/text()'  
xpath["searchLinks"] = '//ol[@class="list-articles"]/li/header//@href'  
xpath["newsArticle"] = '//li//h2/text()'  
xpath["newslinks"] = '//div[@class="g g4"]//header//@href'  
xpath["readHeadlineTitle"] = '//header/h1/span[@class="dh1 head5"]/text()'  
xpath["readHeadlineText"] = '//header/p/text()'  
xpath["readArticleText"] = '//div[@class="formatted"]/p/text()'
```

Spiegel(Site)


```
siteName = "spiegel"  
baseUrl = "https://www.spiegel.de/"  
searchURLString = "https://www.spiegel.de/suche/?suchbegriff="
```



```
xpath = dict()  
[...]
```

Welt(Site)

```
[...]  
[...]
```



Umsetzung: Klassen

- Einfache erweiterbarkeit durch XPath

```
def get_news(self):  
    searchURL = self.baseURL  
    site = requests.get(searchURL)  
    tree = html.fromstring(site.content)  
  
    articles = tree.xpath(self.xpath["newsArticle"])  
    links = tree.xpath(self.xpath["newsLinks"])  
    return articles, links
```

Erweiterung

1. Neue Klasse erstellen

```
class Golem(Site):
    siteName = "golem"
    baseURL = "https://www.golem.de/"
    searchURLString = "https://suche.golem.de/search.php?l=10&q="
    xPath = dict()
    xPath["searchArticle"] = '//span[@class="dh2 head2"]/text()'
    xPath["searchLinks"] = '//ol[@class="list-articles"]/li/header//@href'
    xPath["newsArticle"] = '//li//h2/text()'
    xPath["newsLinks"] = '//div[@class="g g4"]//header//@href'
    xPath["readHeadlineTitle"] = '//header/h1/span[@class="dh1 head5"]/text()'
    xPath["readHeadlineText"] = '//header/p/text()'
    xPath["readArticleText"] = '//div[@class="formatted"]/p/text()'

def get_site_obj(site):
    if site == "golem":
        obj = site2.Golem()
    elif site == "spiegel":
        obj = site2.Spiegel()
    else:
        obj = None
    return obj
```

2. Elif ergänzen

Erweiterung

3. Datentyp erweitern
4. Modell trainieren

```
"types": [  
  {  
    "name": "Site",  
    "values": [  
      {  
        "name": {  
          "value": "spiegel"  
        }  
      },  
      {  
        "name": {  
          "value": "welt"  
        }  
      },  
      {  
        "name": {  
          "value": "zeit"  
        }  
      },  
      {  
        "name": {  
          "value": "golem"  
        }  
      }  
    ]  
  },  
]
```

Deployment (Docker)

```
docker build . -t gst
```

```
docker run -d --rm -p 5001:5000 gst
```

Port 5001 muss auf vorgesehenen Port geändert werden.

Container läuft ohne Output, löscht sich beim stoppen automatisch und leitet Port 5001 auf den Container Port 5000

Demo

https://developer.amazon.com/alexa/console/ask/build/custom/amzn1.ask.skill.c6f4542a-efbd-4272-a3d3-fc73e58277f1/development/de_DE/intents