Finding the Needle in a Haystack

Samrat Baul,

Manager, VUI & Analytics

Voxify

March 2, 2009

The Problem

- Context-based, unstructured search on free-text documents in speech applications is **daunting**
  - Example – Troubleshooting, Catalog Shopping or Frequently Asked Questions
- Menu-driven approaches to such searches are
  - Inefficient
  - Require intensive application design
- SLM approaches are time-consuming and costly

The Problem (contd.)

System: Welcome to the Air Canada Information Line. When you hear the option you want, just say it - Travel Planning, Baggage, Reservations, Checking In, At the -

Caller: Checking In.

System: Ok. Checking In. Remember, you can say Main Menu at any time.

Now, choose from one of these: Policies, Check In Methods, or Check-in Times.

Caller: Methods.

System: OK. There are three ways to check in. Please choose: Web, Mobile, or Airport Kiosk.

Caller: Mobile…

Changing the Approach

- What's on the caller's mind?
  - The caller is looking for ONE **Answer**
  - Predefined Menus hurt efficacy
    - They confuse callers
    - Require more interactions to get to an Answer
- Anatomy of an Answer
    - Synopsys - General carry-on baggage policies
    - Body:
      - Air Canada allows 2 pieces of carry-on luggage, however, departures from any United Kingdom airport allow for one piece only…Neither of these items should weigh more than 10 kilos.

A Proposed Solution

- A classical open-ended question
- Followed by a series of options

System: Welcome to Air Canada Information Line. Please let me know what you're looking for. For example, 'What's my baggage allowance?'

Caller: How many pieces of baggage can I carry on board?

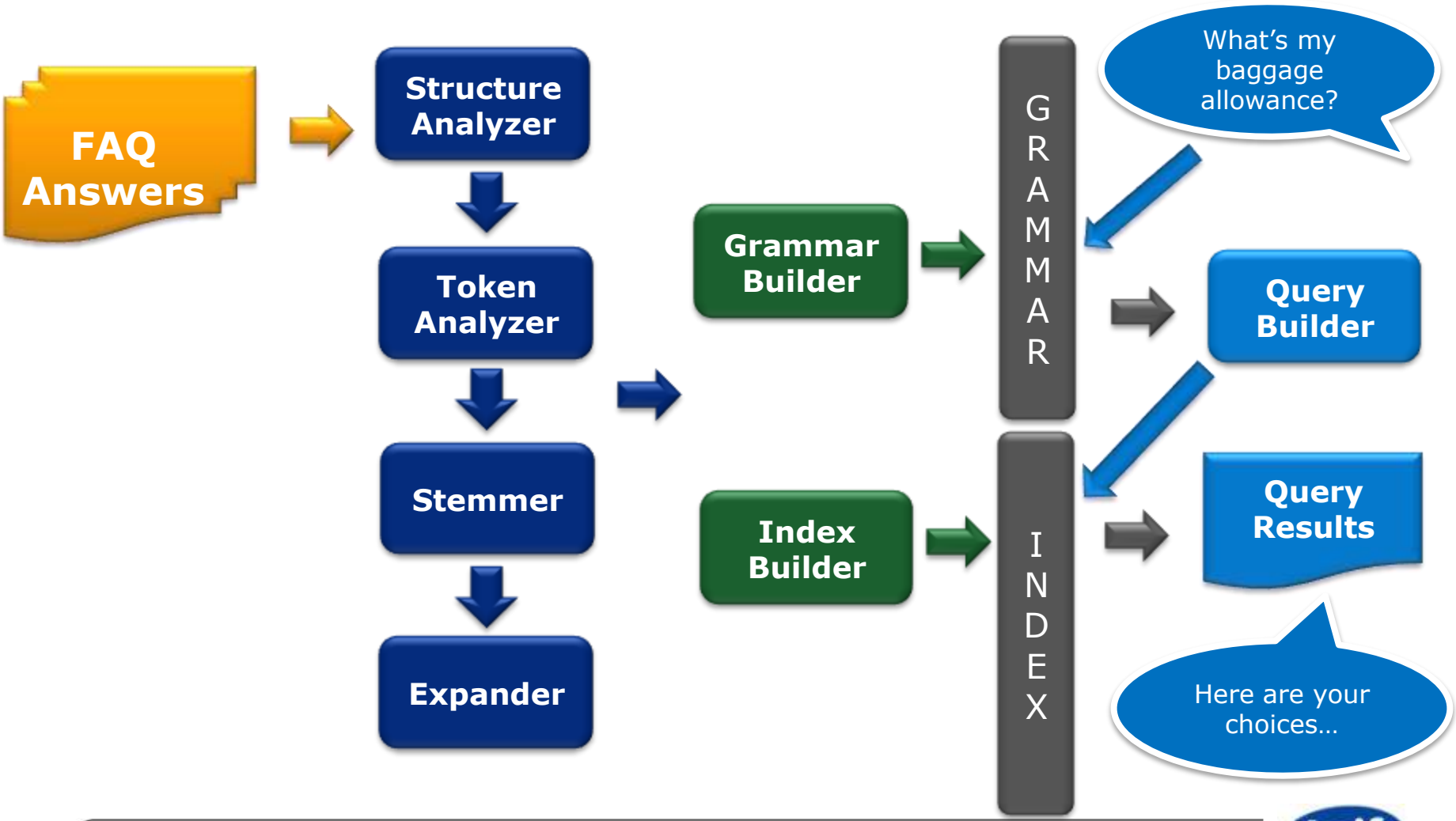
System: Great. I found a few matches for that. Choose the one you want to hear about.

Say 1 for carry on baggage allowance.

Or Say 2 for baggage allowance for children and infants

Caller: "One please"

Architecture



Implementation

- Stemmer
  - Children → Child
- Syn Expander
  - Children → Child → Kid
- Morph Expander
  - Children → Child → Kid → Kids

Implementation (contd.)

- Building the Grammar
  - Tokens + Fillers + Garbage Model + Weight Model
- Building the Query
  - Recognized Tokens – Fillers
- Query Results
  - Ranked List of Documents
    - Synopsys – Used in the Selection UI
    - Body – Played back once Selected

Results

- Improved Satisfaction Rates
  - Lower cognitive load
  - Improved efficacy
- Improved ROI
  - Quick Design & Development
- Reduced call times – Lowering running costs
- Scalable Application Design
  - New documents are easily integrated
  - Robustness

Questions?

Samrat Baul
samrat.baul@voxify.com
www.voxify.com