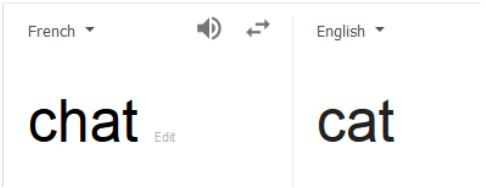
# Using IBM Watson cloud services to build Natural Language Processing solutions to leverage chat tools

Sarah Packowski
spackows@ca.ibm.com

Arun Lakhana
arun.lakhana@gmail.com



# Presentation overview

1. Context
2. Challenge
3. Solution
4. Results
5. Lessons learned

# Context

– where we're coming from

- By “Chat tools”, we mean:

  *Any tool that allows web page users to communicate in real time, or with some delay, with the owners of the web page in a little chat window on the web page. There might be none, some, or complete integration with an automated “bot”.*

- My team started using IBM Watson NLP cloud services to process our chat convos:
  - “Real-time” apps for filtering, routing, answering
  - Analysis of historical convos

- Other teams saw what we were doing and asked us for help

# Challenge

- Some teams receive ~1000 chat convos/week
- Some teams were investing a great deal of effort to “understand” users' needs: analyzing metrics, surveys, interviews, customer advocate programs... while leaving thousands of historical messages from real users untapped

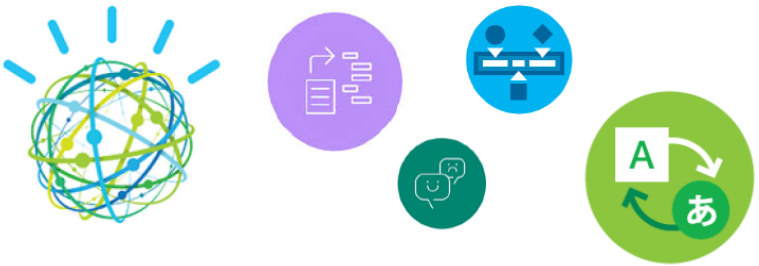
# Solution

## Strategy: Divide and conquer

- Handle messages in languages we can't support
- Filter “noisy” messages
- Classify messages into buckets (eg. “Account-related”, or “Problem” vs. “Question”)
- Extract key idea from messages
- Automatically answer “easy” questions

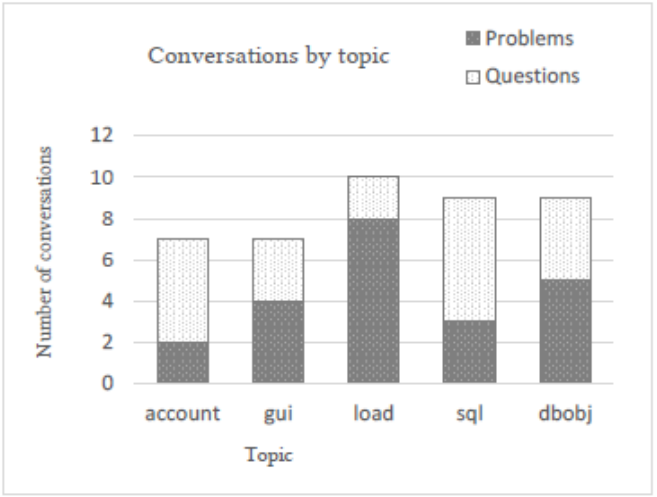
## Tools: IBM Watson cloud services

- Watson Language Translator
- Watson Natural Language Classifier
- Watson Knowledge Studio
- Watson Natural Language Understanding
- Watson Tone Analyzer

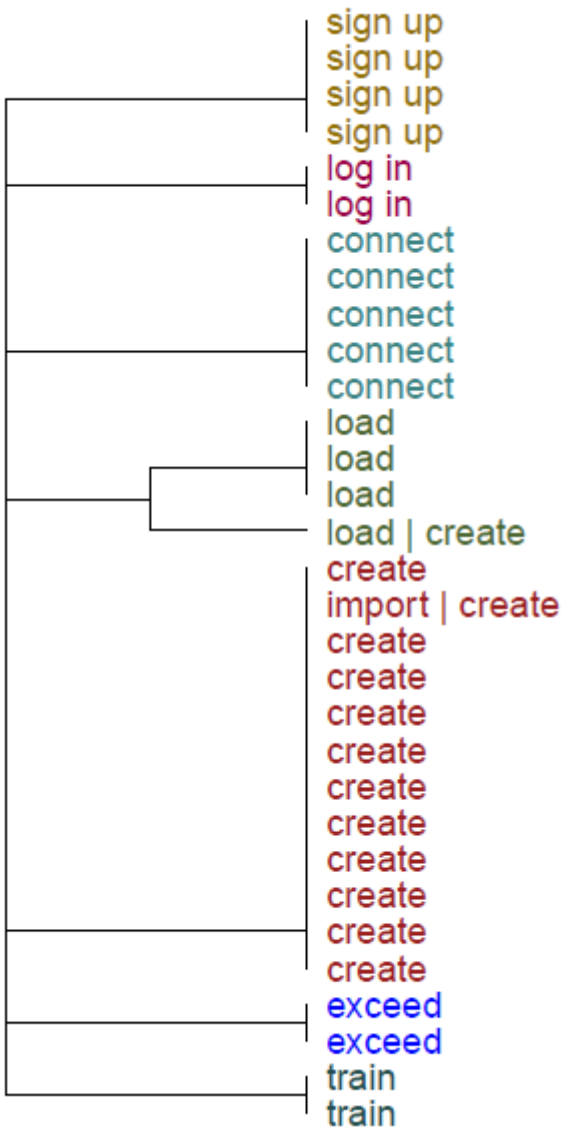


# Results

- Reducing Support team load by 30%
- Insight into what our users are saying: pain points, FAQs, wish lists ...







# Lessons learned

1. How do people use chat?
2. How much training data is needed?
3. Is a trained cognitive component irreplaceable?
4. Can you reuse a cognitive component?
5. Are these cognitive tools useful for every problem?

Note: All but #3 are really lessons learned about integrating AI/cognitive into standard software development processes – lessons about human assumptions and misunderstandings about AI, not technology ones

# How do people chat?

In bursts.

- We needed to capture users' *first complete idea*, not just the first text they submit

Hi

Can I ask a question?

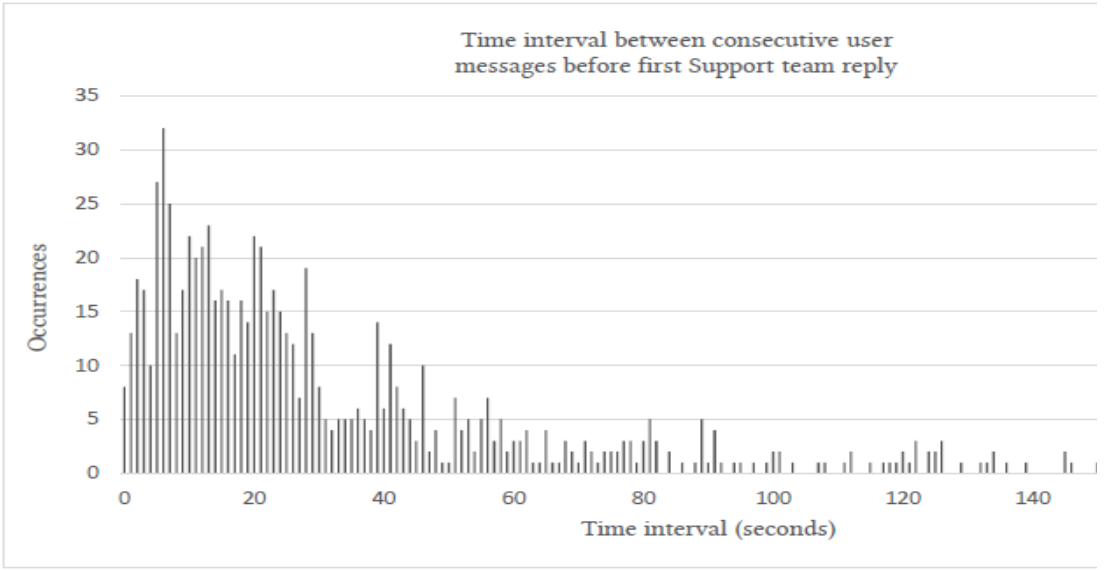
Yes! We're here to help

How do I upload a file?

Hi

I'm using Feature X

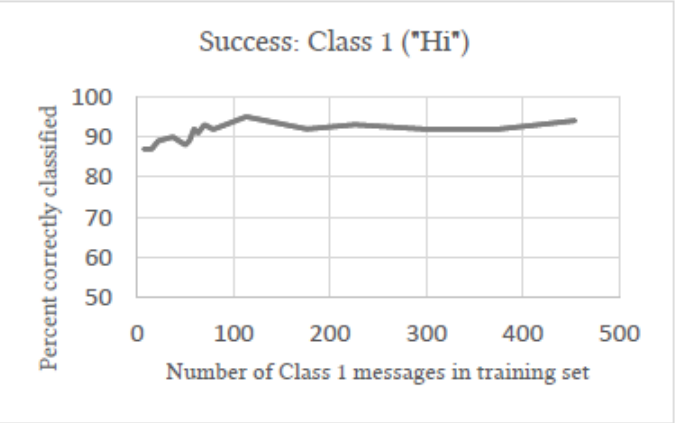
How do I upload a file?

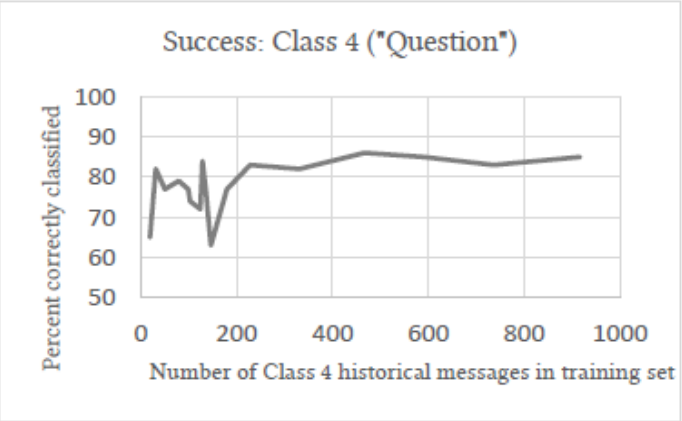


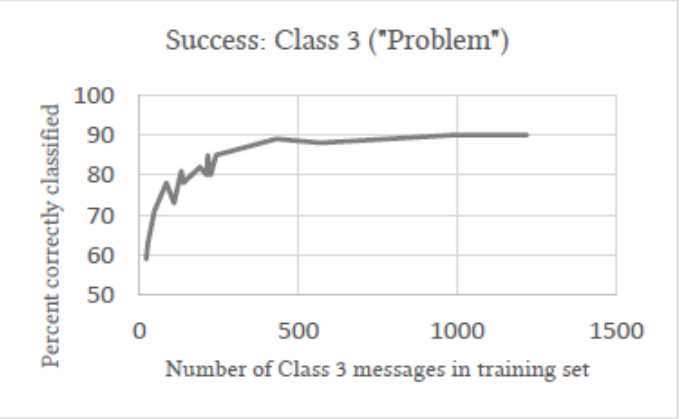
# How much training data is needed?

More is better, *to a point*.

- Collecting training data can be expensive
- Automated tests tell you when you're good enough



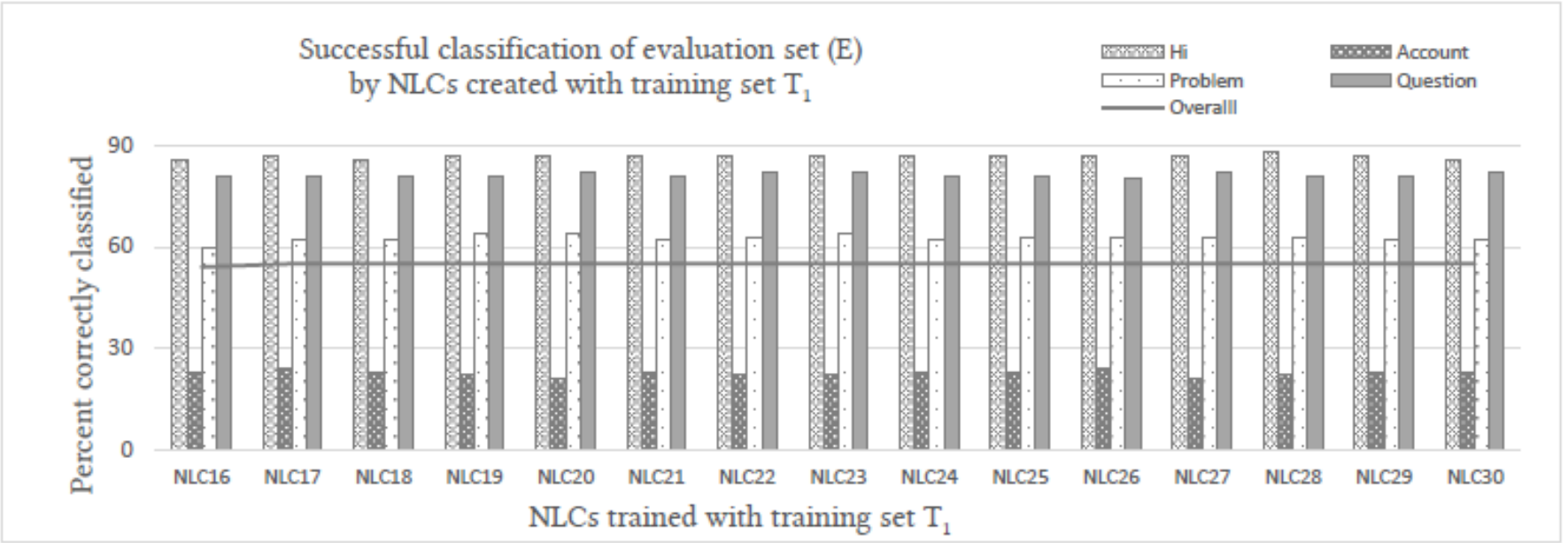



# Is a trained cognitive component irreplaceable?

We could reliably reproduce classifier results.

- People view AI and cognitive as fuzzy, black boxes, but these services are reliable
- Save the training data, not the trained component



# Can you reuse cognitive components?

Don't assume that you can.

- Re-use is such a strong aspect of software development that every project manager has asked if we could re-use an existing solution
- We haven't seen good-enough results from re-using existing cognitive pieces
- Generalization: don't assume “out-of-the-box” or pre-trained cognitive solutions will work for your case

# *A little bit wrong is... totally useless

General language model
- Too much noise
- Missed most important

Hello, could You provide an output of following command executed on the serverr? db2 list utilities show details
Source

db2 list utilities | output | command | serverr | details

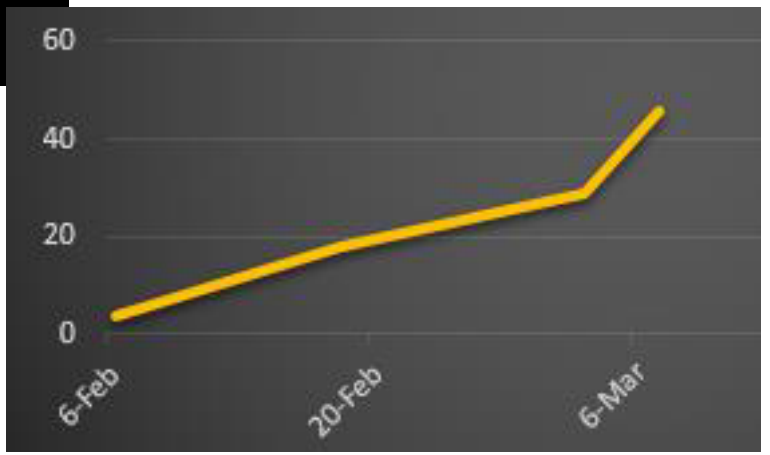
hello, is there any chance to execute command db2 list utilities show details for our instance? We are loading a huge amount of data for 24 hours and do not know where we are :(

command db2 list | instance

Custom language model
- Just right

```
Entities:
[
   {
      "type": "COMMAND",
      "text": "db2 list utilities",
      "count": 1
   }
]
```

```
Entities:
[
   {
      "type": "COMMAND",
      "text": "db2 list utilities",
      "count": 1
   },
   {
      "type": "ACTION",
      "text": "loading",
      "count": 1
   }
]
```

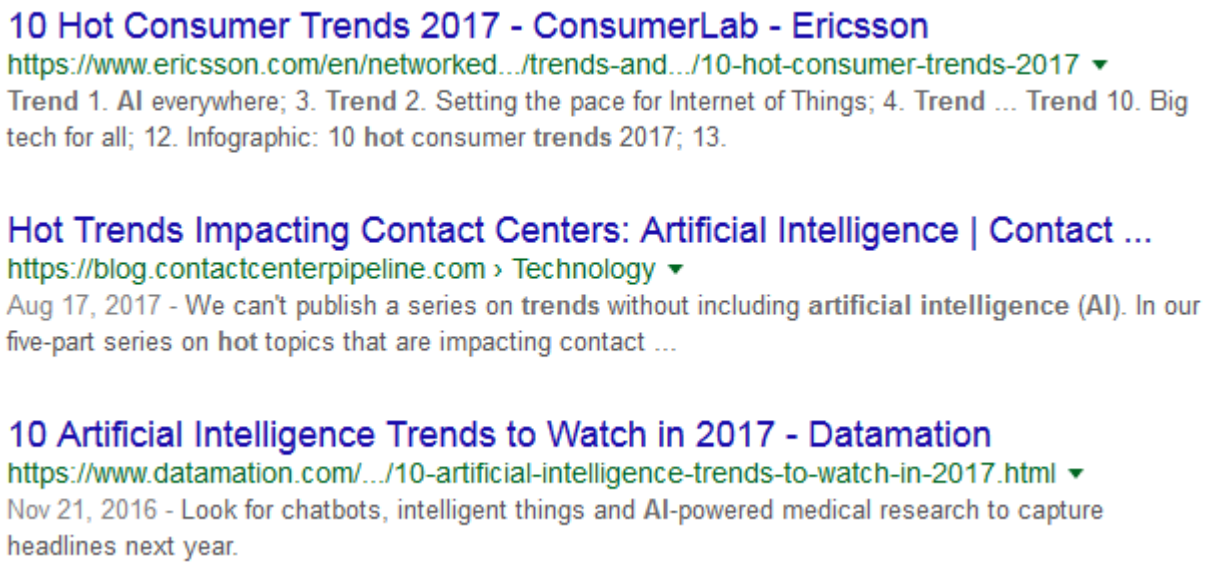



➔ You still need good, old-fashioned software engineering practices, such as automated regression testing..

# Can I use cognitive for... everything?

No.

- AI/cognitive is the hot thing, so a lot of people want to apply it to everything
- Start simple, add cognitive where it fits



# Try it for yourself!

Build a cognitive app that processes text comments.

http://ibm.biz/CASCON-2017-Sample-Code

- Sample data
- Sample Code
- Sample output
- Instructions – including video!

**Drop by demo booth D6 tomorrow to see this in more detail**