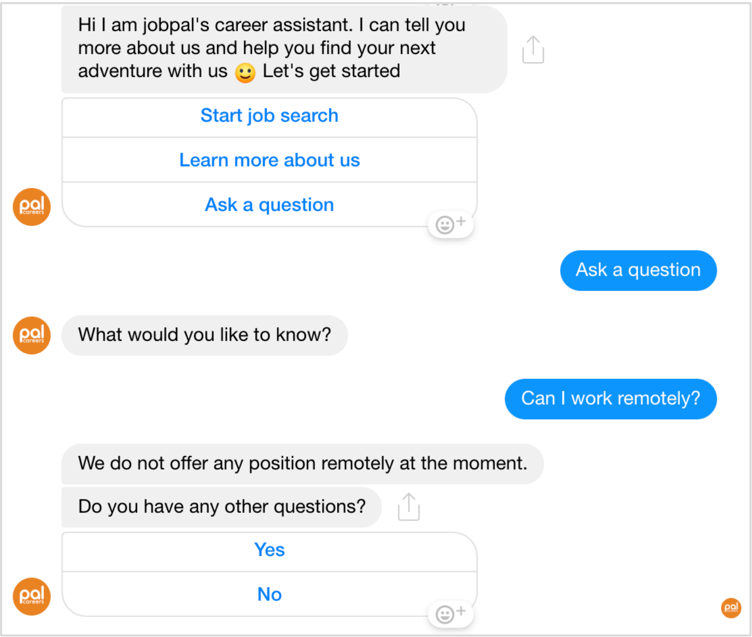
Introducing `nex-cv`

- a metric for estimating multi-class classifier performance based on cross-validation
- adapted for improvement of small, unbalanced natural-language datasets used in chatbot design
- uses negative examples in the evaluation of text classification



Hi I am jobpal's career assistant. I can tell you more about us and help you find your next adventure with us 🙂 Let's get started

Ask a question

What would you like to know?

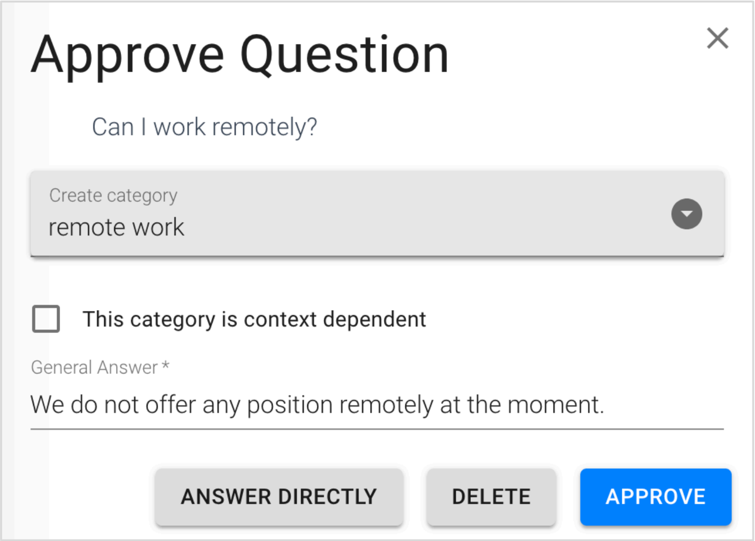
Can I work remotely?

We do not offer any position remotely at the moment.

Do you have any other questions?

What the end-user of a chatbot sees

Our experiences draw upon building **recruitment chatbots** that mediate communication between job-seekers and recruiters by **exposing the ML/NLP dataset to the recruiting team.**



Approve Question

Can I work remotely?

Create category: remote work

□ This category is context dependent

General Answer *: We do not offer any position remotely at the moment.

What the recruiter sees in a dashboard

Evaluation approaches must be be understandable to various stakeholders, and useful for improving chatbot performance. We validate the metric based on seven recruitment domain datasets in English and German over the course of one year.

Evaluation and Improvement of Chatbot Text Classification Data Quality Using Plausible Negative Examples

Kit Kuksenok (`kit@jobpal.ai`) & Andriy Martyniv (`andriy@jobpal.ai`)

ACL 2019 Workshop on NLP for Conversational AI, August 1, 2019

Code: `https://github.com/jobpal/nex-cv`

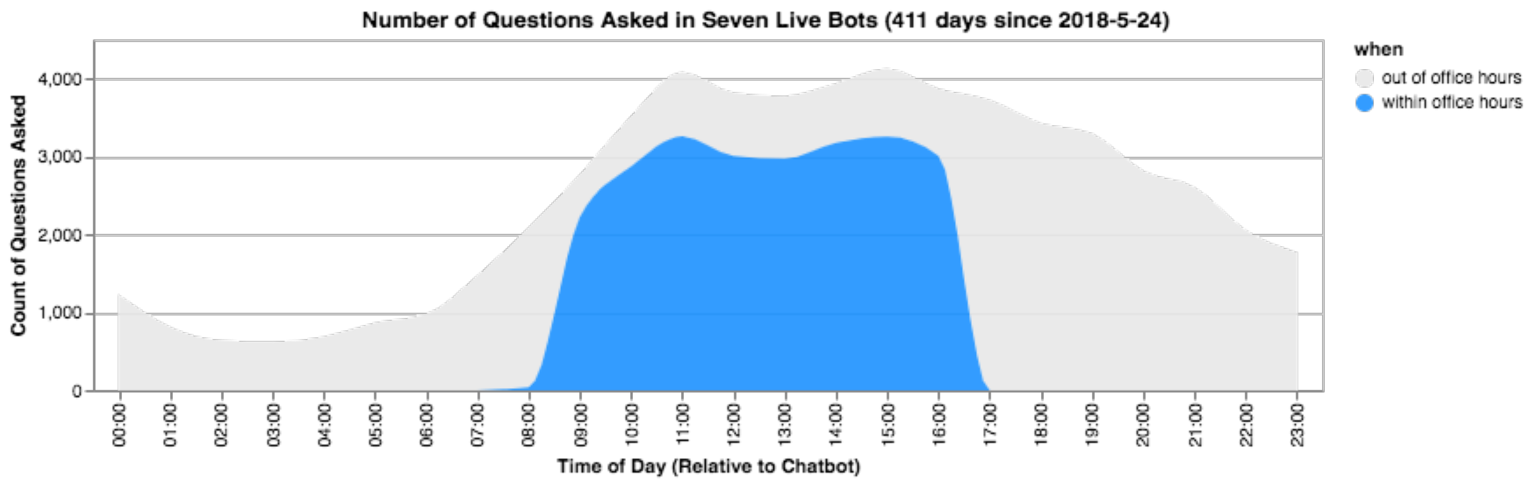
jobpal

Domain: Recruitment Chatbots

- recruiter teams motivated by scale and accessibility to build and maintain chatbots that provide answers to frequently asked questions (FAQs) based on ML/NLP datasets

- enterprise clients may have up to 100K employees, and commensurate hiring rate

- **over 50%** of end-user (job-seeker) traffic occurs outside of working hours or during holidays (consistent with the anecdotal reports that using the chatbot reduces email and ticket load)



25% of end-user queries fall into 5 major themes

1. Application Process
2. Salary
3. Professional Growth & Development
4. Internships
5. Contact a Human

25% of end-user queries fall into 14 minor themes

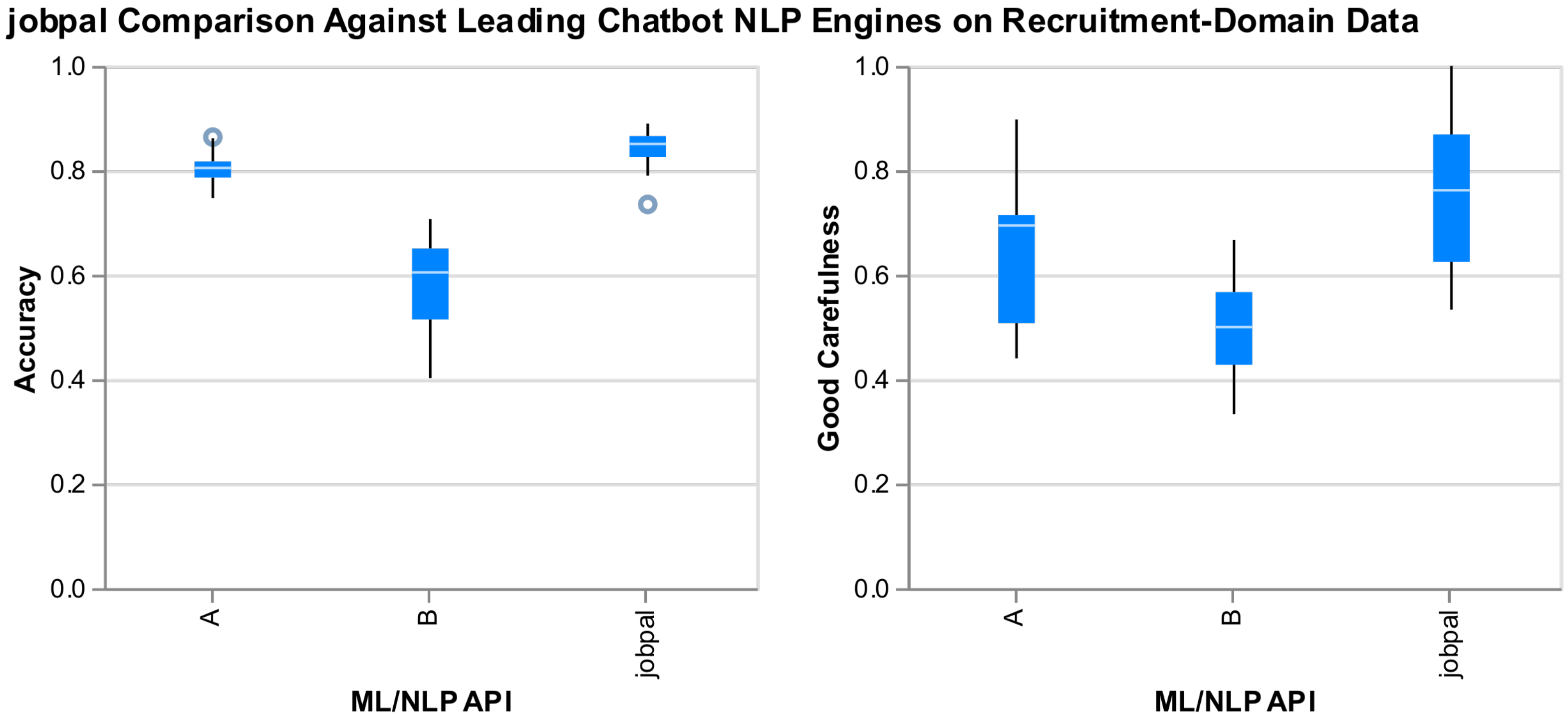
Application Evaluation; Application Deadline; Application Delete or Modify; How Long to Apply and Hear Back; Qualification; Application Documents; Language Expectations; Thesis; Working Hours; Location; Starting at the Company; Commute; Equipment; Benefits.

40% concern company-specific FAQs

The clean, anonymized recruitment-domain-specific dataset in English was built by anonymizing and aggregating all FAQ datasets. It was also used for the experiment comparing performance to other systems.

Validation: Metric Can be Used for Internal and External Comparison

A well-performing chatbot has an automated response rate 70-80% for FAQs; the rest should be effective escalation for niche questions or emerging topics.



- used the nex–cv metric, aggregating three settings of the metric [with (K, P) as (0, 0), (0, 0.15), and (5, 0)] to provide a plausible range of estimated performance.
- metric must account for **fallback or escalation**, which may beimplemented in different ways: as a separate class, or by relying on confidence scores from classifiers that produce measures of confidence
- "carefulness" score represents how useful the confidence score is for **deciding when to decline an answer**: the number of incorrect guesses rejected due to too-low confidence scores divided by total no-answer-given cases

Evaluation and Improvement of Chatbot Text Classification Data Quality Using Plausible Negative Examples

Kit Kuksenok (kit@jobpal.ai) & Andriy Martyniv (andriy@jobpal.ai)

ACL 2019 Workshop on NLP for Conversational AI, August 1, 2019
Code: https://github.com/jobpal/nex-cv

jobpal

Method

Treat low-population classes as **sources of plausible negative examples.**
Low-population classes identified either by cutoff or proportion.

Data Shape

- training set: between 1K and 12K training examples across between 100 and 200 classes
- starting with about 50–70 classes and creating new classes after the system goes live and new, unanticipated user needs are encountered
- low-population classes are typically rare or relatively recent topics, which justifies interpreting them as plausible negative examples

Classification Stack

- spaCy (Honnibal and Montani, 2017)
- fastText (Bojanowski et al., 2016) for vectorization, with transformation for improved performance (Arora et al., 2016)
- logistic regression with L2 regularization (Pedregosa et al., 2011)

Use & Test nex-cv

- Online code provides an abstract black-box definition for a classifier, and two test strategies:
- Integration / consistency testing with `CustomClassifier.test()`
- Functional testing: runing `nex-cv` both `K = 0` and `P = 0` yields comparable results to 5-fold CV.

```
Result: $(X_{train}, y_{train}, X_{test}, y_{test})$
Require data $X, y$ s.t. $x_i$ is the input text that
  has gold standard label $y_i$ $\forall i$;
Require label sets $L_{SM}, L_{LG}$ s.t.
  $L_{SM} \cup L_{LG} = \{y_i \mid y\}$ Require test fraction
  $0 < t < 1$ and function $split_t(L)$ which
  randomly splits out two lists $L_1, L2$ s.t.
  $\frac{|L_2|}{|L|} = t$ and $L_1 \cup L_2 = L$ ;
for $L_j \in L_{LG}$ do
    $TR, TS = split_t(i|y_i \in y \wedge y_i == L)$;
    $X_{train}, y_{train} \leftarrow x_i, y_i$ s.t. $i \in TR$ ;
    $TR, TS = split_t(i|y_i \in y \wedge y_i == L)$;
    $X_{test}, y_{test} \leftarrow x_i, y_i$ s.t. $i \in TS$ ;
end
$TR_L, TS_L = split_t(\{j|y_j \in L_{SM}\})$;
$X_{train}, y_{train} \leftarrow x_i, y_i$ s.t. $y_i \in TR_L$;
$X_{test}, y_{test} \leftarrow x_i, \emptyset$ s.t. $y_i \in TS_L$;
```

Algorithm 1: Negative Example Data Provision

```
Result: $L_{SM}, L_{LG}$
Require data $X, y$ s.t. $x_i$ is the input text that
  has gold standard label $y_i$ $\forall i$;
Require cutoff parameter $K > 0$ ;
$L_{SM} = \{y_i \mid y_i$ in $y$, occurs $< K\}$ ;
$L_{LG} = \{y_i \mid y_i$ in $y$, occurs $\geq K\}$ ;
```

Algorithm 2: Cutoff Selection of Plausible Negative Example Classes

```
Result: $L_{SM}, L_{LG}$
Require data $X, y$ s.t. $x_i$ is the input text that
  has gold standard label $y_i$ $\forall i$;
Require proportion parameter $0 \leq P < 1$ ;
$L_{SM} = \{\}$ ;
Let $Q = \{y_i \mid y_i \in y\}$, as queue sorted from
  least to most occurring in $X$ ;
while $\frac{|\{i|x_i \in X \wedge y_i \in L_{SM}\}|}{|X|} < P$ do
    Pop element $L$ from $Q$ ;
    $L_{SM} \leftarrow L$;
end
$L_{LG} = \{y_i \mid y_i$ in $y$, not in $L_{SM}\}$ ;
```

Algorithm 3: Proportional selection of Plausible Negative Example Classes

Evaluation and Improvement of Chatbot Text Classification Data Quality Using Plausible Negative Examples

Kit Kuksenok (kit@jobpal.ai) & Andriy Martyniv (andriy@jobpal.ai)

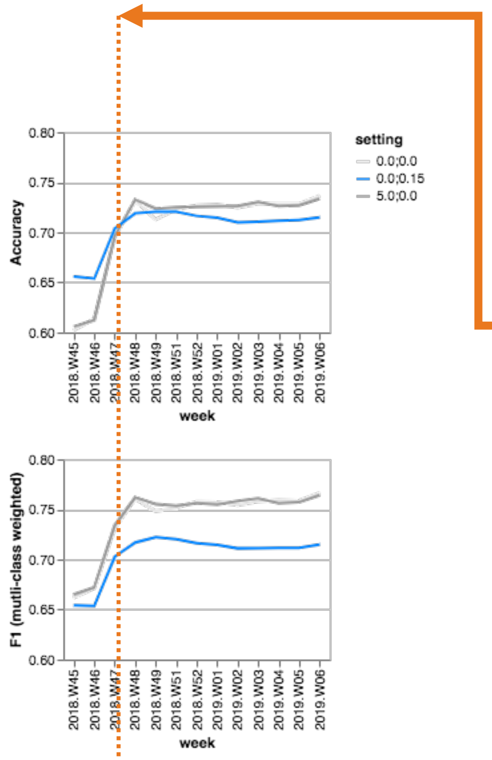
ACL 2019 Workshop on NLP for Conversational AI, August 1, 2019
Code: https://github.com/jobpal/nex-cv

jobpal

Validation: Metric Enables Data Quality Improvements

Existing chatbot guidelines include "**transparency**" as an important topic, but, in practice, why something does not work, and under what conditions, can puzzle designers and developers, not just end-users. The `nex-cv` metric:

- produces accuracy scores more in line with human judgment than e.g., CV + F1 (details in paper)
- ensures that data from low-population classes is included in both training and testing, at least as a "negative example"
- can be used to generate internal **recommendations** and as part of ongoing data quality maintenance



Candidates for Merging

Recommendations: The principle of this analysis is to solve the worst problems first. The most common problem we have is when a smaller category overlaps in meaning with a larger one, and degrades both. Below are the suggested pairs to focus on.

The category **Remote_work** has fewer questions (only 15)
than **Company_location** (which has 26).
This may mean that Remote_work should be merged into Company_location; redistributed elsewhere in the dataset; or enriched and refined.

Internal tools use the metric to provide **actionable guidance** that has helped improve overall data quality significantly and consistently. The above recommendation is a generated text summary of confusion matrix results that is used by non-developer staff to improve data quality.

Evaluation and Improvement of Chatbot Text Classification Data Quality Using Plausible Negative Examples

Kit Kuksenok (kit@jobpal.ai) & Andriy Martyniv (andriy@jobpal.ai)

ACL 2019 Workshop on NLP for Conversational AI, August 1, 2019
Code: https://github.com/jobpal/nex-cv

Background: Data Quality and Maintenance in Chatbots

Classes — “intents” — are **trained with synthetic data and constitute anticipated use**, rather than actual use. Existing general chatbot platforms include this synthetic data step as part of design and maintenance.

For example, when it comes to invocations for a voice agent (Ali et al., 2018), dataset construction encodes findings about how users imagine asking for an action: the authors use crowdsourcing to achieve both consistency useful for classification, and **reflection of user expectations** in the dataset.

We work on enabling domain-experts (recruiters) to maintain the dataset, which helps map end-user (jobseeker) needs to recruiters’ goals.

Data cleaning is not only relevant to chatbots. Model-agnostic systems for understanding machine learning can help iteratively develop machine learning models (Zhang et al., 2019).

Feature engineering can be made accessible to non-developers or domain experts, e.g. (Ribeiro et al., 2016). We make use of representative examples in the process that surfaces nex-cv to non-developers; using the the **inspection-explanation-refinement** approach employed in (Zhang et al., 2019). Enabling non-developers to perform data cleaning effectively allows developers to focus on model adjustments and feature engineering.

There are many ways to measure overall chatbot quality, such as manual check-lists of high-level feature presence (Kuligowska, 2015; Pereira and D´ıaz, 2018).

User behavior measurements— both explicit, like ratings or feedback, and implicit, like timing or sentiment— are explored in (Hung et al., 2009).

During metric development, we used qualitative feedback from domain-expert users, and key performance indicators (KPIs), such as automatic response rate. The use of **a classifier as a component in a complex flow** demands robust and actionable evaluation of that component.

Evaluation and Improvement of Chatbot Text Classification Data Quality Using Plausible Negative Examples

Kit Kuksenok (kit@jobpal.ai) & Andriy Martyniv (andriy@jobpal.ai)

ACL 2019 Workshop on NLP for Conversational AI, August 1, 2019

Code: https://github.com/jobpal/nex-cv

jobpal