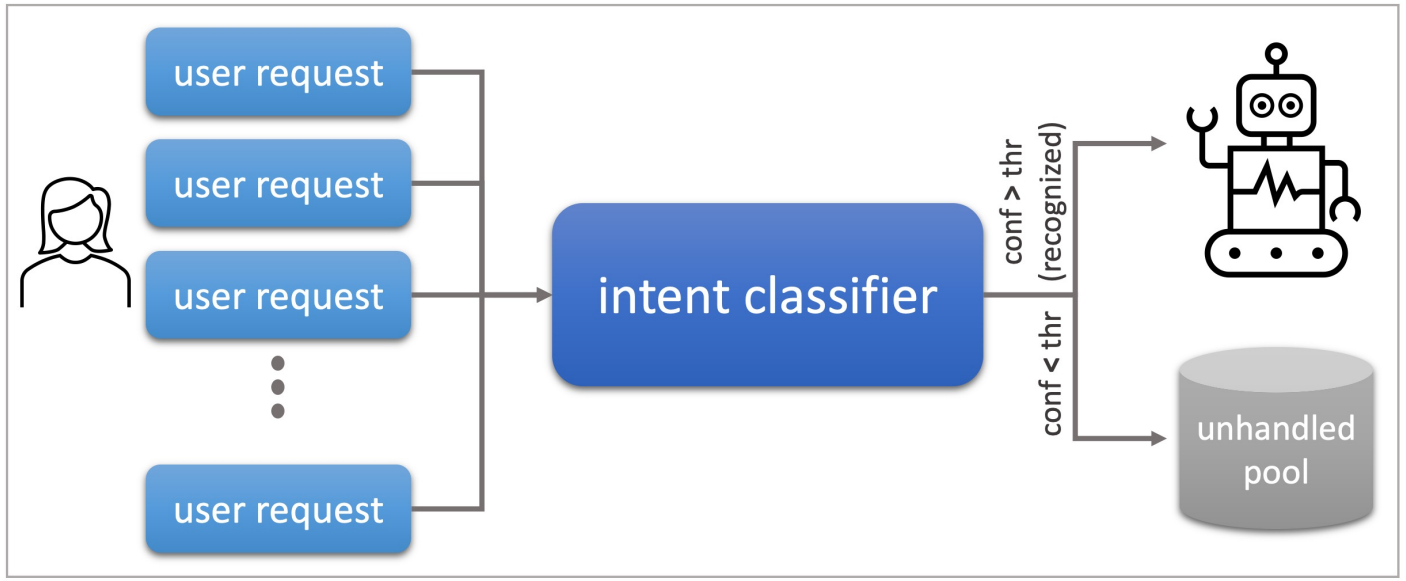
Analysis of Unrecognized User Requests in Goal-Oriented Dialog Systems

the final project

4 Jan 2023

MOTIVATION AND BACKGROUND

- Goal-oriented dialog systems, a.k.a virtual assistants (VAs), often fail to recognize the intent of natural language requests
- In practice, these cases are normally identified using intent classifier uncertainty – requests that are predicted to have a level of confidence below a certain threshold are reported as unrecognized (or unhandled)

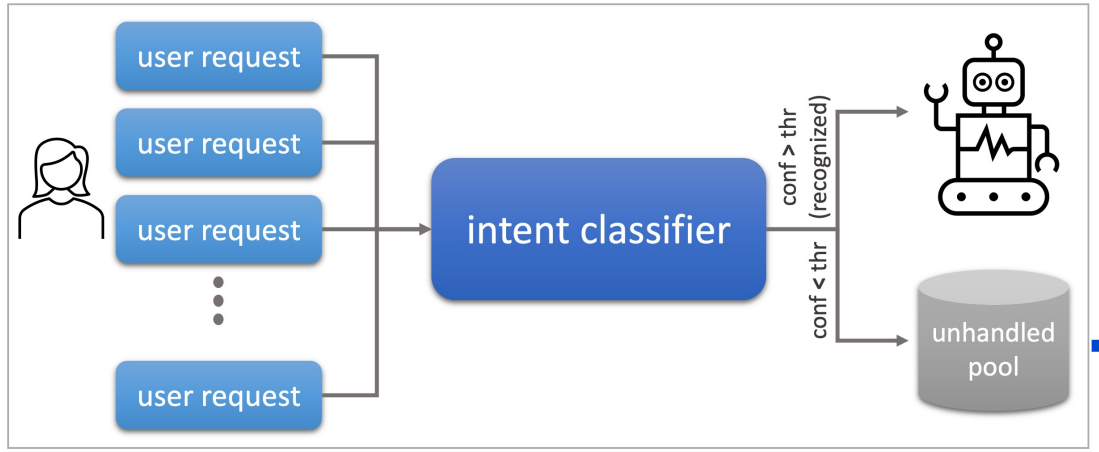


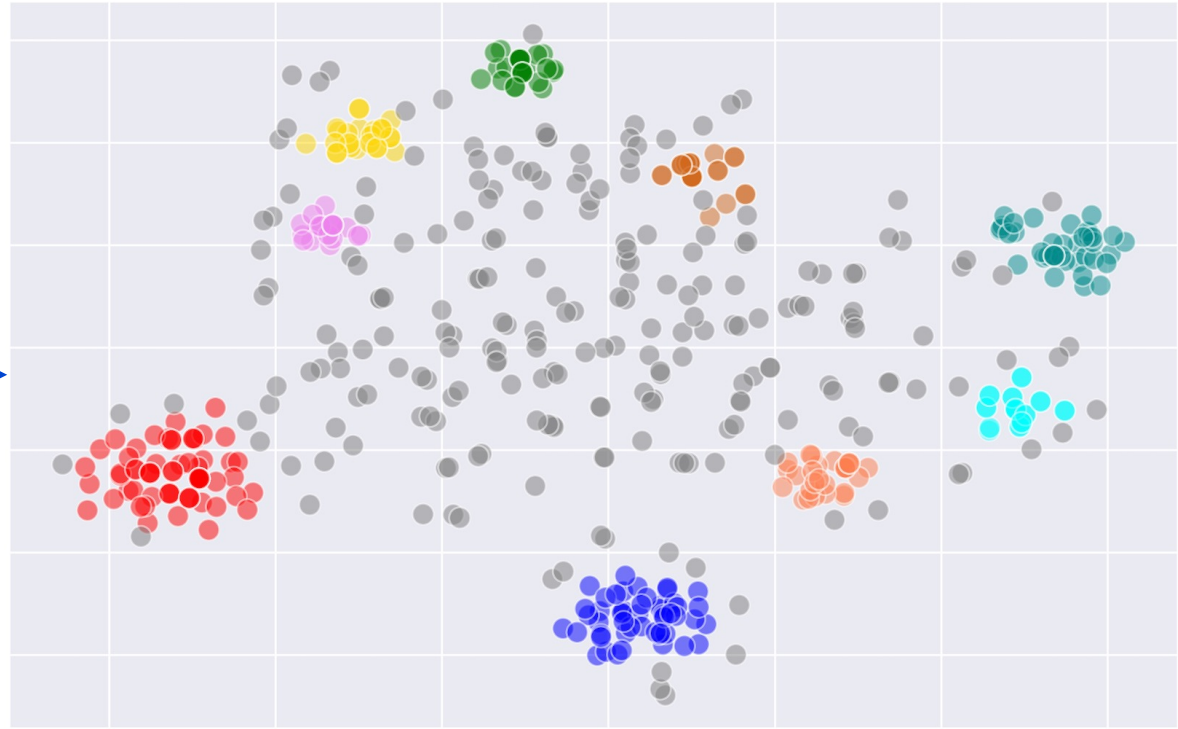
MOTIVATION AND BACKGROUND

- Unhandled requests carry over various aspects of potential importance
  - novel examples of existing intents, completely novel topics, seasonal peaks
- In large deployments the number of unhandled requests can reach tens of thousands daily, making manual inspection impractical

- Our goal is to propose (and implement) an approach for
- (1) surfacing topical clusters in unhandled requests (clustering)
- (2) extraction of cluster representatives
- (3) cluster naming (labeling)

(1) CLUSTERING REQUESTS





what properties should the clustering approach satisfy?

(1) CLUSTERING REQUESTS – requirements

Clustering solutions can be roughly categorized into across two major dimensions:

✗	requiring a predefined (fixed) number of output clusters	vs	discovering the number of clusters as part of the clustering algorithm	✓
✗	forcing cluster assignment on the entire dataset	vs	tolerating outliers	✓
	for instance, the KMeans clustering		required in our use-case	

(1) CLUSTERING REQUESTS – example

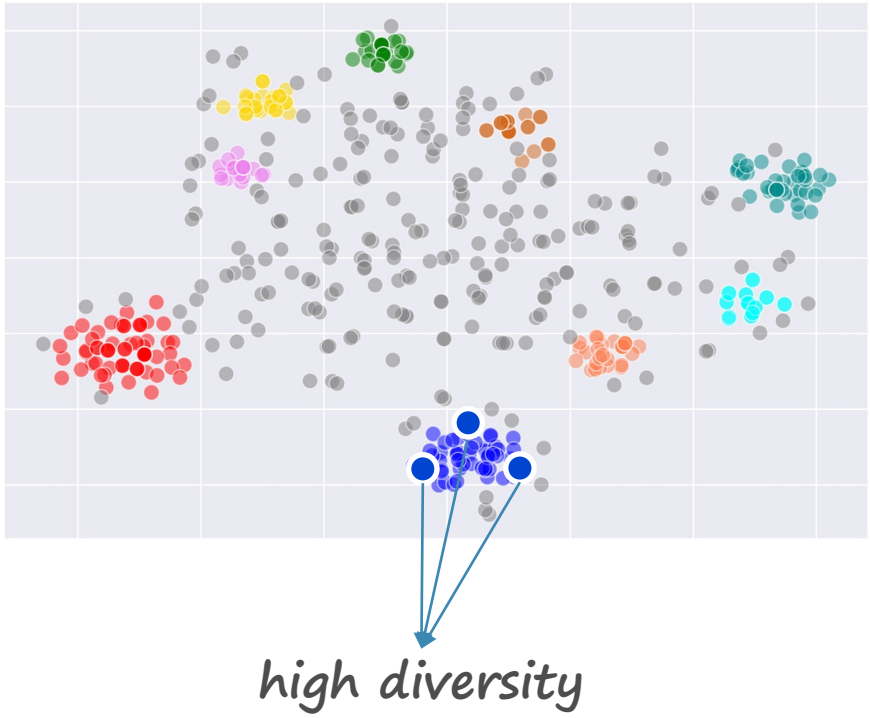
cluster name: **difference covid flu** (28)	cluster name: **covid during pregnancy** (17)
is covid the same as flu? (4)	covid 19 and pregnancy (6)
how is covid different from the flu? (3)	covid risk for a pregnant woman (4)
what's the difference between covid 19 and flu?	what is the risk of covid for pregnant women?
what's the difference between covid and flu	is covid-19 dangerous when pregnant?
is the covid the same as cold?	7 months pregnant and tested positive for covid, any risks?
covid vs flu vs sars	covid 19 during pregnancy
…	…

(1) CLUSTERING REQUESTS – algorithm suggestion

- encode a set of m unhandled requests: R=(r1, r2, r3, …) into vector representations (embeddings): E=(e1, e2, e3, …)
- iterate over representations in E, where each request can be assigned to an existing cluster (if its proximity to the cluster's centroid meets some similarity threshold), otherwise the request initiates its own cluster
- additional iterations over all request embeddings are performed till the algo convergence or till the max number of iterations is exhausted
- clusters with size exceeding a pre-defined min_size are considered as generated clusters
  - all other requests are considered unclustered

(2) EXTRACTION OF CLUSTER REPRESENTATIVES

- large (or immature) VA deployments may introduce tens of thousands of unrecognized requests daily → large clusters, often impractical for manual processing
- for each cluster, we would like to select K cluster representatives satisfying the property of diversity:
  - **diversity** of a request set mirrors differences in the various ways people express the same need



(3) CLUSTER NAMING (LABELING)

- essential step for better consumability of the results
- a plausible cluster name should be a well-formed word n-gram (fluency) and reflect the n-gram frequency in a cluster (faithfulness)*
  - (7) new contact (ADJ NOUN)
  - (9) create new contact (VERB ADJ NOUN)
  - (4) email address (NOUN NOUN)

* you can think about other possible ways to achieve the same goal

EVALUATION

- (1) surfacing topical clusters in unhandled requests (clustering)
- multiple dataset(s) with (good) clustering solution will be provided as a ground truth
- quantitative evaluation

- (2) extraction of cluster representatives
- example set of extracted cluster representatives will be provided
- qualitative evaluation

- (3) cluster naming (labeling)
- example assignment of cluster names will be provided
- qualitative (+ possibly quantitative) evaluation

PROJECT SUBMISSION – a single zip file with

- a report (3-4 pages) including the description of
  - your approach to the task (the three parts), examples and analysis
  - evaluation of the clustering outcome against the provided solution – RI, ARI
    - user the provided `compare_clustering_solutions.py` to compute these scores
  - any essential details about running your code (e.g., anticipated runtime)
- your outcome on the datasets attached to the project – two output json files
  - the output files should be precisely in the same json format as the provided solutions
- your code (the `main.py` file)

- project submission due date: Feb 20th