MSA Project Week 2024

Team Analytical Midtowners

Divij Mishra
Manikant Thatipalli
Shiven Barbare

Contents

Problem statement

Your challenge: train a text classifier to mimic the output of a zero-shot classification from a foundation model

Challenges

1. Heavy imbalance in dataset

Top 5 labels (all > 10k):

product details inquiry	42698
product availability and stock	37972
schedule repair	35386
change or update order	24297
defective product	19269

Bottom 5 labels (all < 100):

payment failed	601
account cancellation	520
reschedule order pickup	478
performance issues	473
network or connectivity issues	409

Challenges

1. Heavy imbalance in dataset
2. Lots of similar labels

```
('product details inquiry', 'product compatibility')
('defective product', 'damaged product')
('schedule order pickup', 'reschedule order pickup')
('schedule delivery', 'reschedule delivery')
('schedule repair', 'reschedule repair')
('lost or forgot items', 'delivery of parts of delivery items missing')
('account security', 'login issues', 'forgot my password')
```

Challenges

1. Heavy imbalance in dataset
2. Lots of similar labels
3. Dirty data

aaaa ??
ab ??
aback ??
abaco ??
abandon
abata ??
abbey ??
abbishnew
abbot
abbreviate

jdi ??
jealous
jean
jeanette
jeanie ??
jeannette
jeannie
jeans
jeben ??
jeep

press
pressed
presser
pressio ??
pressort ??
presssssssss ?????
pressure
pressway
preston
prestron ??

Challenges

1. Heavy imbalance in dataset
2. Lots of similar labels
3. Dirty data
4. Need to limit model complexity

Solutions

1. Heavy imbalance in dataset **- label clusters** + **cascading models**
2. Lots of similar labels **- label clusters** + **cascading models**
3. Dirty data
4. Need to limit model complexity

Solutions

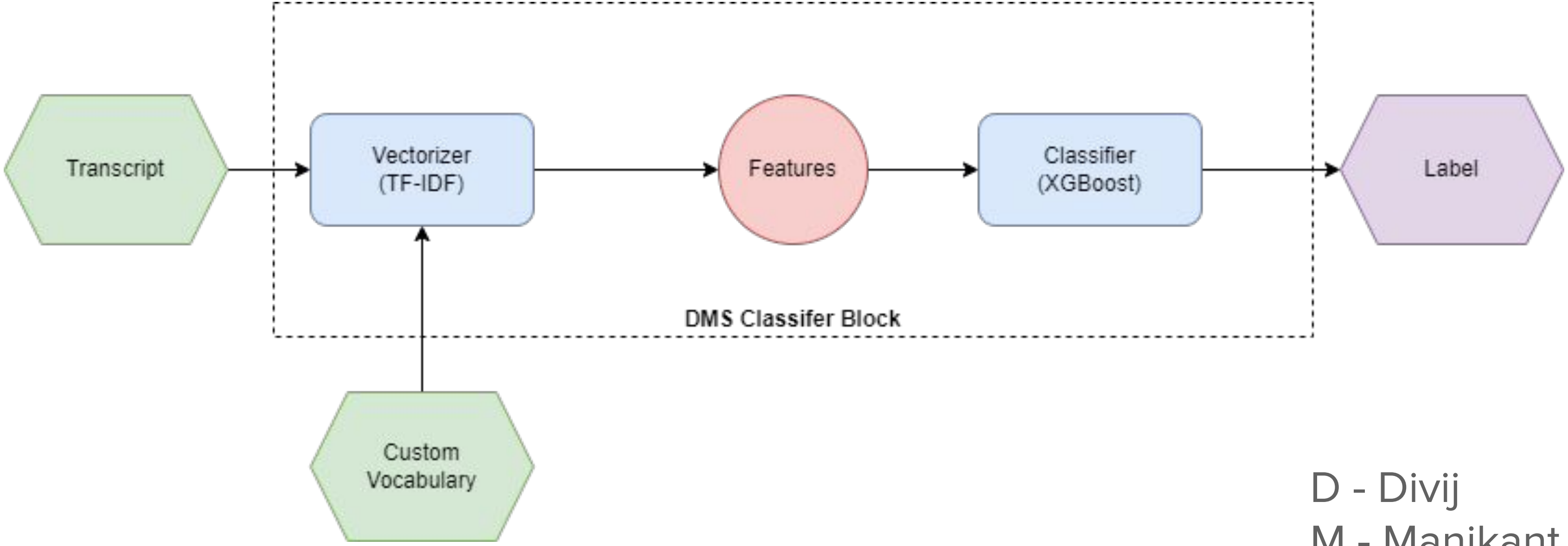
1. Heavy imbalance in dataset **- label clusters** + **cascading models**
2. Lots of similar labels **- label clusters** + **cascading models**
3. Dirty data **- custom vocabulary**
4. Need to limit model complexity

Solutions

1. Heavy imbalance in dataset **- label clusters** + **cascading models**
2. Lots of similar labels **- label clusters** + **cascading models**
3. Dirty data **- custom vocabulary**
4. Need to limit model complexity **- TF-IDF** + **XGBoost**


LLMs with billions of parameters
This is brilliant.
TF-IDF, XGBoost
But I like this.
imgflip.com

Model design 1: DMSClassifier block



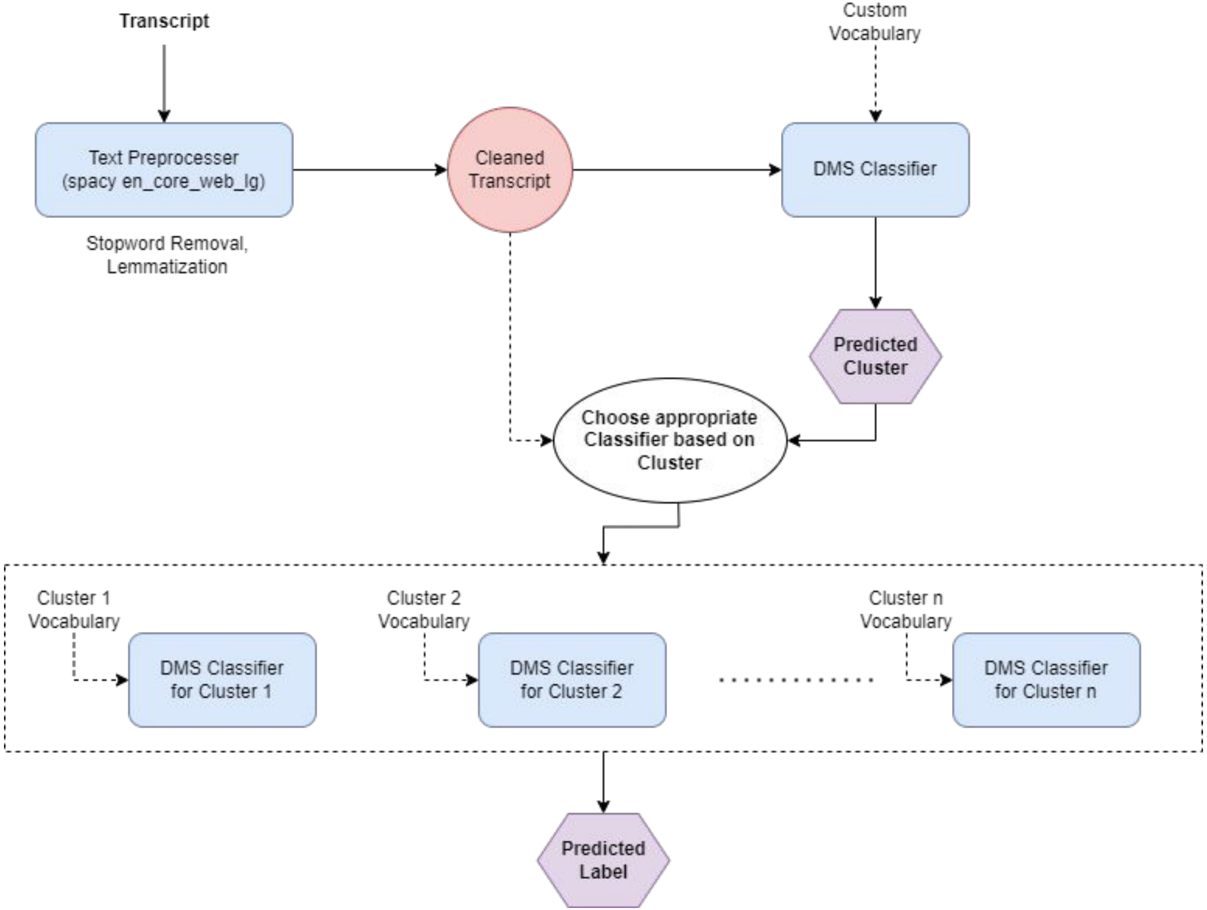
D - Divij
M - Manikant
S - Shiven

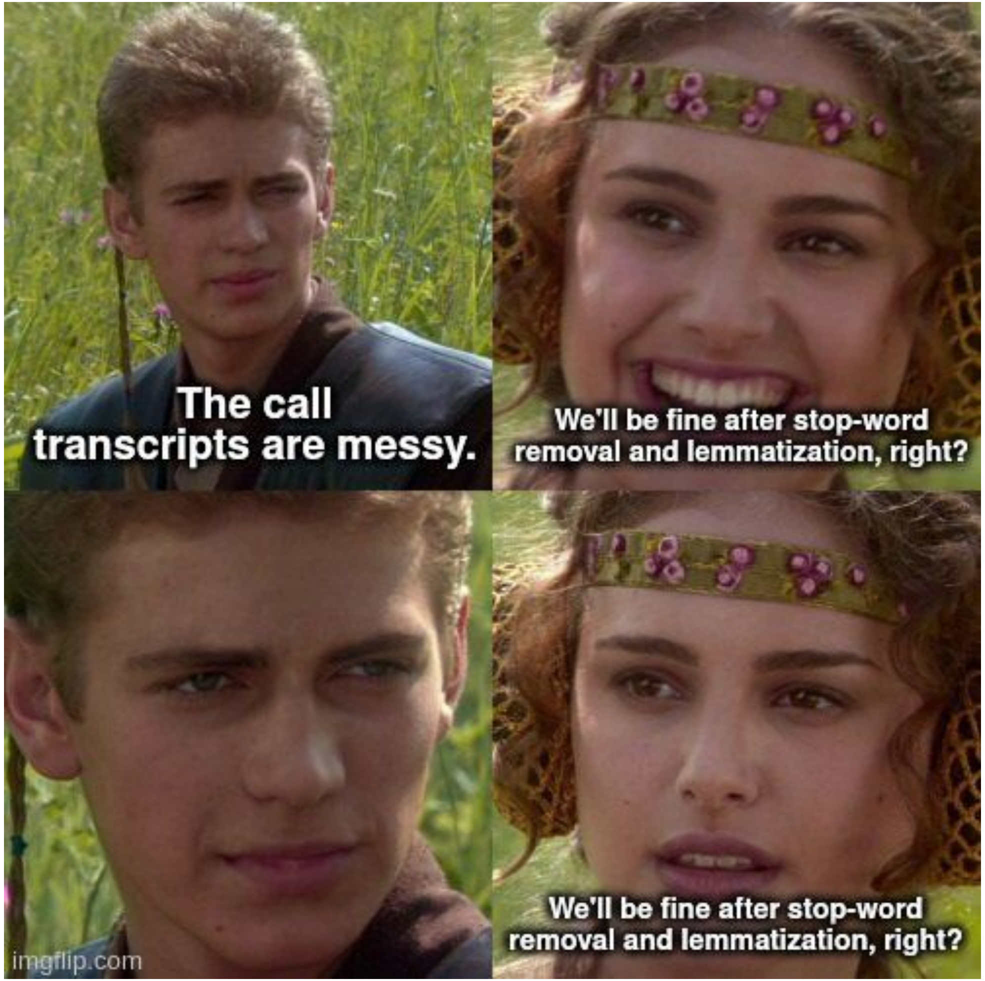

You already have a classifier.
We have one, yes. But
what about a second classifier?
imgflip.com

Model design 2: Cascade pipeline

Every transcript goes through a cascade of 2 DMSClassifiers:

1) The first one predicts the cluster
2) The second one predicts the label




The call
transcripts are messy.
We'll be fine after stop-word
removal and lemmatization, right?
imgflip.com
We'll be fine after stop-word
removal and lemmatization, right?

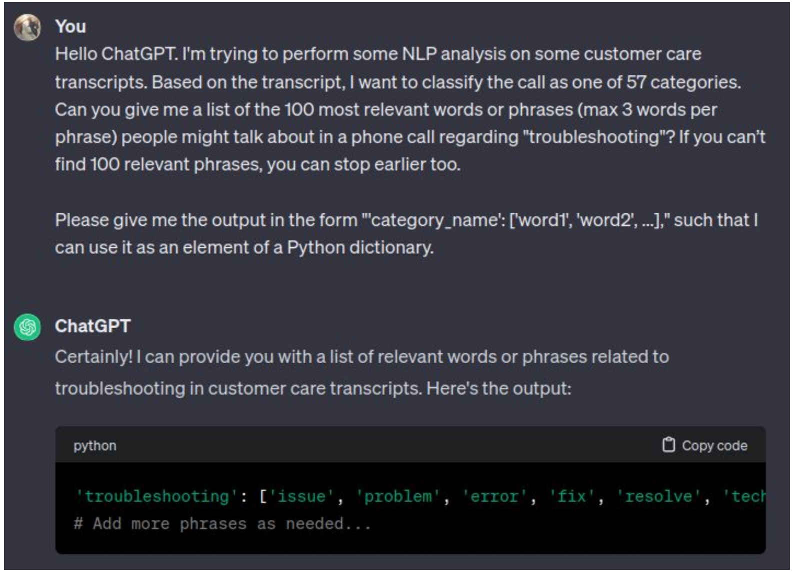
Vocabulary generation

1. Still have 25k words after stop-word removal, lemmatization + TF-IDF (many of these are either not words, or not useful words)

aaaa ??	jdi ??	press
ab ??	jealous	pressed
aback ??	jean	presser
abaco ??	jeanette	pressio ??
abandon	jeanie ??	pressort ??
abata ??	jeannette	pressssssss ?????
abbey ??	jeannie	pressure
abbishnew	jeans	pressway
abbot	jeben ??	preston
abbreviate	jeep	prestron ??

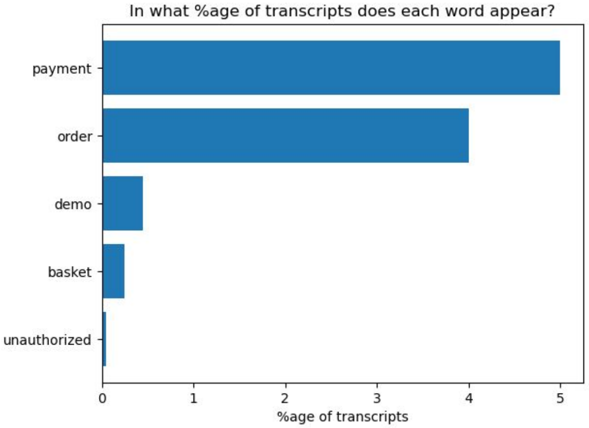
Vocabulary generation

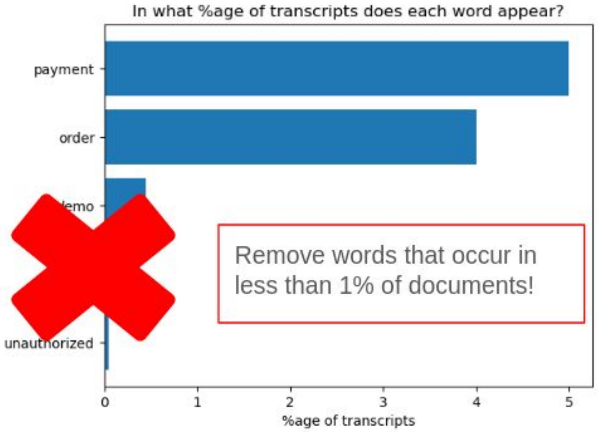
1. Still have 25k words after stop-word removal, lemmatization + TF-IDF (many of these are either not words, or not useful words)
2. **Solution:** Use ChatGPT to generate lists of relevant words! E.g.



Vocabulary generation

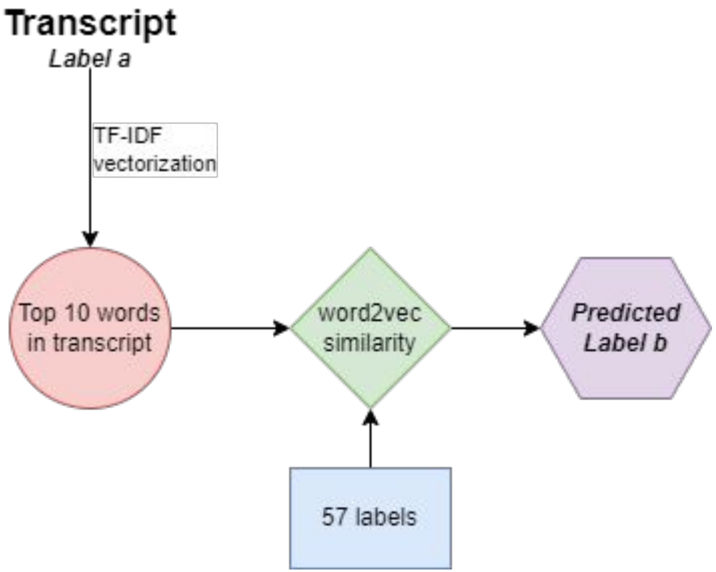
1. Still have 25k words after stop-word removal, lemmatization + TF-IDF (many of these are either not words, or not useful words)
2. **Solution:** Use ChatGPT to generate lists of relevant words!
3. We got 5k words from ChatGPT - reduced this to roughly 400





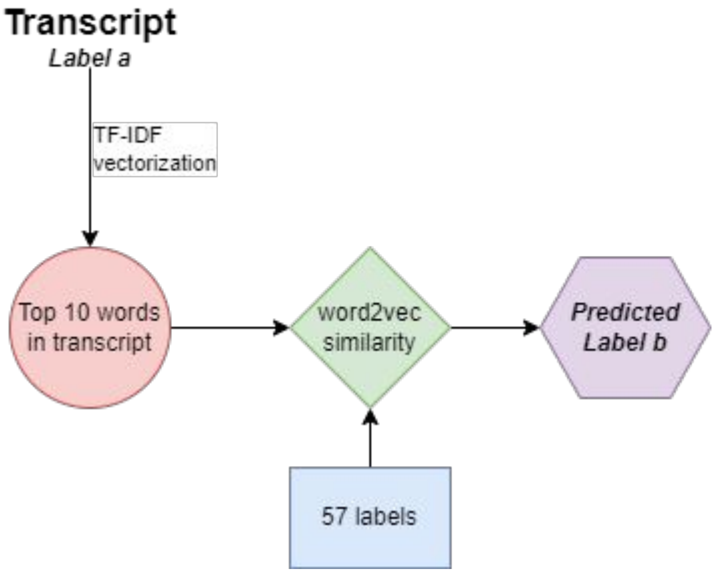
Clustering

1) Basic model for finding clusters



Clustering

1) Basic model for finding clusters
2) Used the table as a heuristic to manually engineer meaningful clusters



Actual(right) Predicted (below)	payment method	change payment method	login issues
payment method	29	10	2
change payment method	12	20	1
login issues	1	3	28

Clustering

```
A='change payment method'
top10_words_list_A
'card': 31,
'payment': 22,
'order': 18,
'credit': 16,
'credit card': 15
'update': 11,
'good': 11,
'number': 9,
'go': 9,
'change': 8,
```

```
B='payment method'
top10_words_list_B
'card': 35,
'payment': 25,
'credit': 25,
'credit card': 23
'good': 14,
'order': 11,
'date': 9,
'number': 9,
'renew': 7,
```

```
C='login issues'
top10_words_list_C
'account': 24,
'email': 21,
'password': 18,
'phone': 15,
'number': 14,
'try': 14,
'log': 13,
'time': 9,
'code': 8,
'reset': 6,
```

Similar Labels

Different Label

Clustering

Final clusters:

```
cluster_label
order related and payments    126932
warranty                      124098
product queries               104195
queries regarding website      15504
authorization                   8599
```

New label - “other”

Still have heavy imbalance within clusters - introduced a label “other” for these

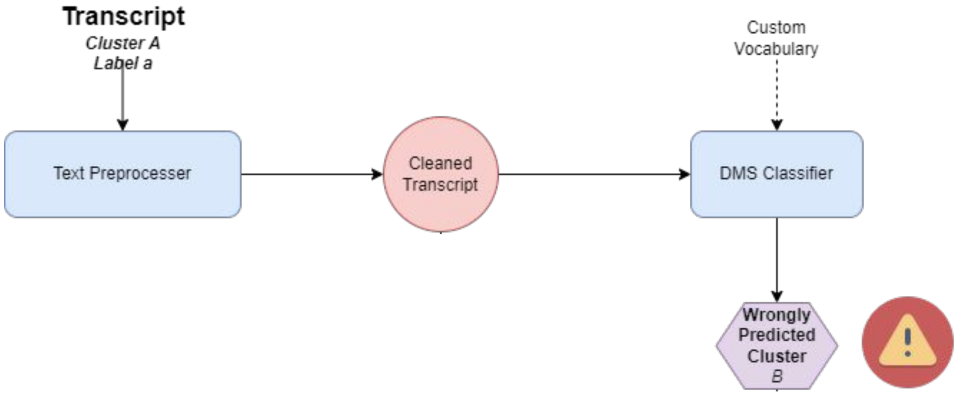
Cluster: Warranty

schedule repair	35386
defective product	19269
schedule installation	14616
troubleshooting	14242
damaged product	9065
software error	7502
software installation	6156
reschedule repair	3759
warranty claim	3006
screen issues	2706
device damaged	2230
check warranty coverage	2166
lost or forgot items	2028
reschedule installation	1494
performance issues	473

Clubbed together into label “other”

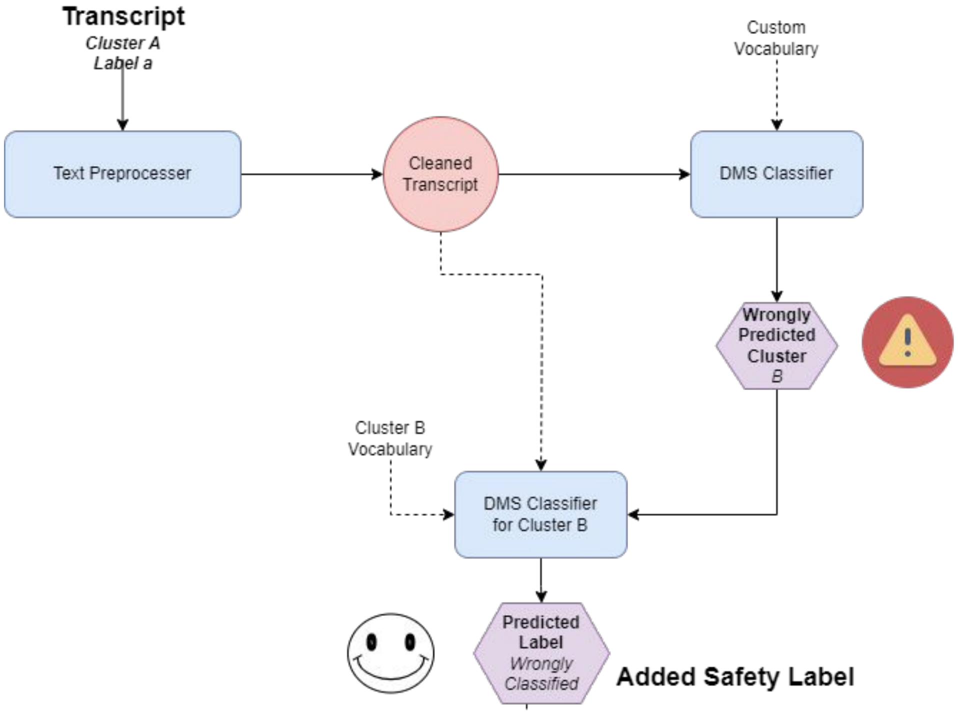
New label - “wrong cluster”

1) Errors due to incorrect cluster mapping result in noisy inputs for the cluster level classifier



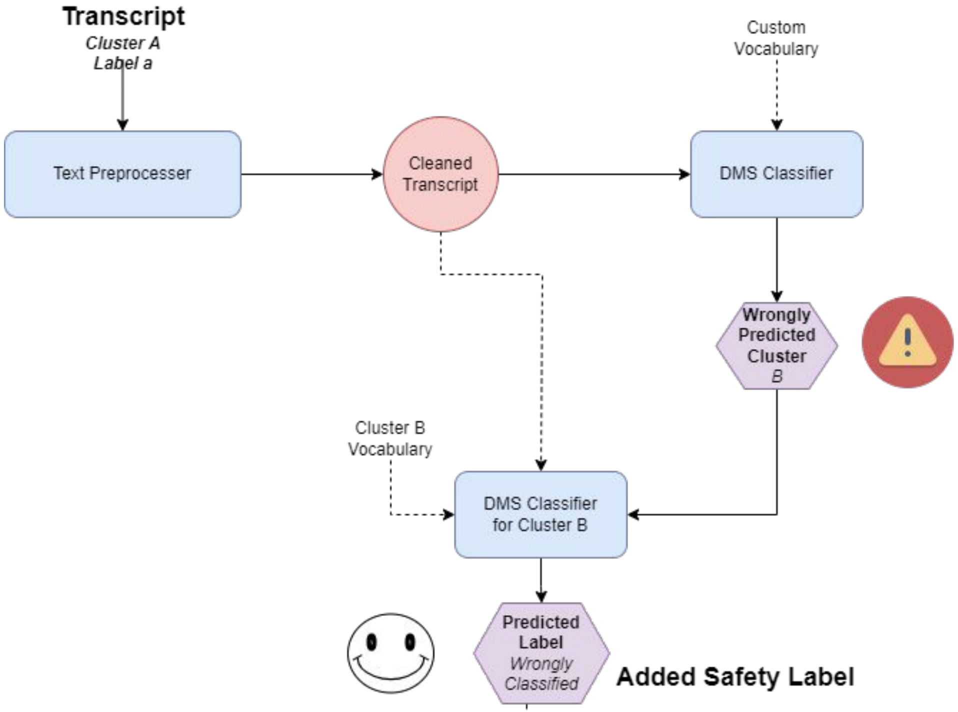
New label - “wrong cluster”

1) Errors due to incorrect cluster mapping result in noisy inputs for the cluster level classifier



New label - “wrong cluster”

1) Errors due to incorrect cluster mapping result in noisy inputs for the cluster level classifier
2) To train this, each cluster level classifier got 10% data from other clusters



Metrics

1. Classification metrics

Classifier	F1-score
Fine-tuned T5 small	0.72
Our pipeline	0.50
Clustering	0.76
"Authorization"	0.38
"Order"	0.53
"Product"	0.54
"Queries regarding website"	0.36
"Warranty"	0.46

Metrics

1. Classification metrics

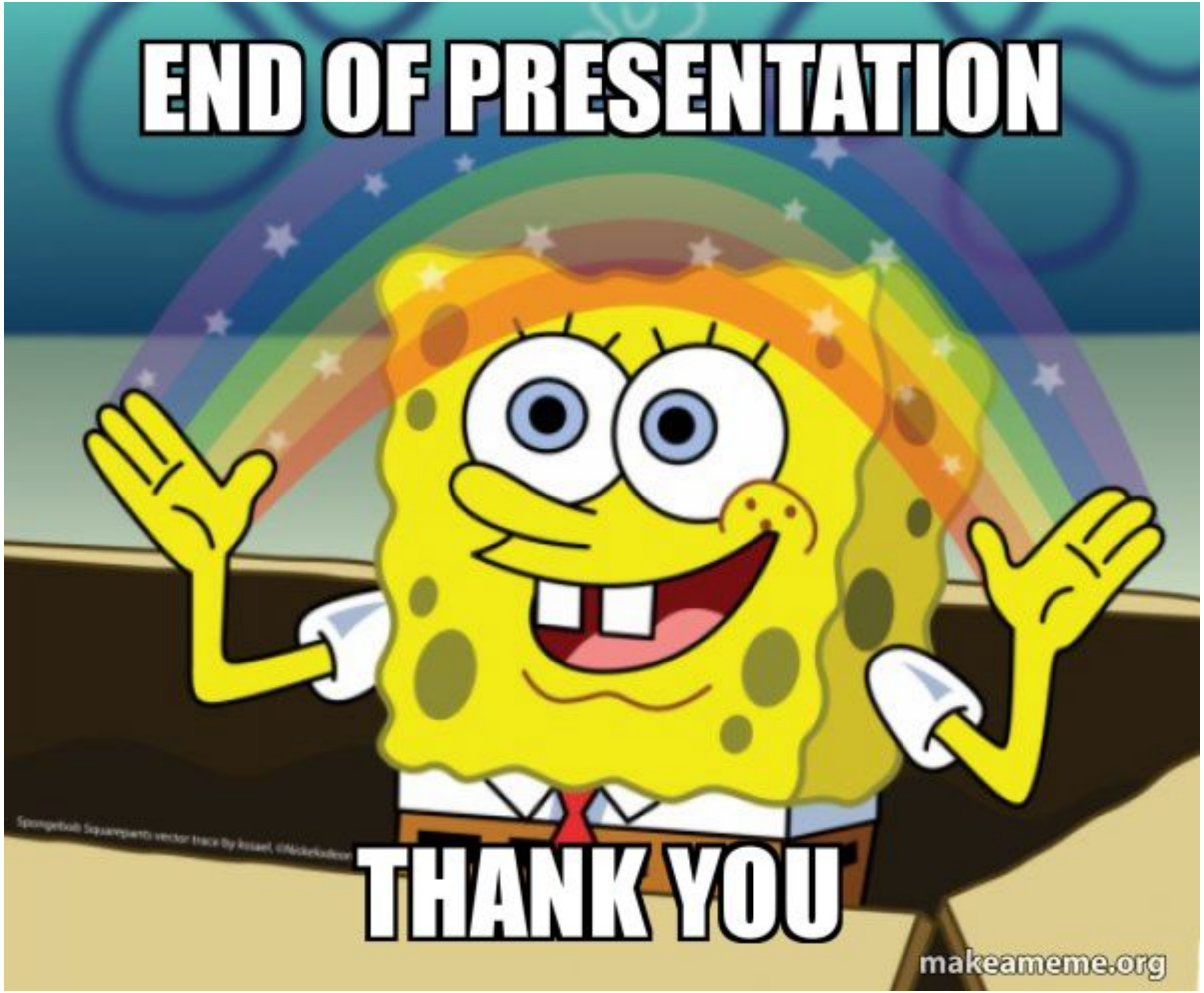
Classifier	F1-score
Fine-tuned T5 small	0.72
Our pipeline	0.50
Clustering	0.76
"Authorization"	0.38
"Order"	0.53
"Product"	0.54
"Queries regarding website"	0.36
"Warranty"	0.46

2. Things needed for inference:
   a. spaCy en_core_web_lg
   b. TF-IDF vectorizer trained on 350k
   c. Custom vocabularies
   d. 6 XGBoost models (each inference only requires 2)

Lightweight model (relative to big LLMs)!

Limitations and further scope

1. Limited time and compute resources -> better hyperparameter tuning could improve performance
2. Limited experimentation in choosing clusters and vocabulary
3. Decided to do 2 cascade layers because of time constraints -> could create further classifiers for minority classes in “other” category
4. Could use SHAP for better explainability


END OF PRESENTATION
THANK YOU
makeameme.org