

Online Appendix

APPENDIX A FEATURES

for
“Tell Me More! Using Multiple Features for Binary
Text Classification with a Zero-Shot Model”

by

David Broneske, Nikhilkumar Italya, & Fabian
Mierisch

TABLE I: Features Waiting List

Yes - Email is a waiting list offer	No - Email is not a waiting list offer
Wir können Sie auf die Warteliste setzen	Wir können Sie auf keine Warteliste setzen
Die Aufnahme ist möglich	Die Aufnahme ist nicht möglich
Die Plätze sind noch nicht voll	Die Warteliste ist voll
Bitte tragen Sie sich ein	Bitte eintragen Sie sich nicht
Bitte registrieren Sie sich	Bitte registrieren Sie sich nicht
Bitte melden Sie sich an	Bitte anmelden Sie nicht an
Sie können sich einschreiben	Sie können sich nicht einschreiben
Eventuell haben wir noch Platz	es tut mir leid
die Zulassung ist noch offen	Die Registrierungen sind abgeschlossen
wenn Sie wollen	Wir müssen Ihre Anfrage abweisen
Bitte füllen Sie das Formular aus	
Bitte benutzen Sie das Portal	
Bitte benutzen Sie den Link	
zurückschicken	
Bitte füllen Sie alle Angaben aus	

Notes: Table shows the features for outcome classes “Yes” and “No” of the outcome *Waiting List*. Features aim to provide context for the classification of the algorithm while simultaneously comply with class definitions. Out of the 12’547 observations 9’976 (79.5%) were humanly labeled as a waiting list offer and 2’571 (20.49%) as not a waiting list offer.

TABLE II: Features Answer

Yes - Email contains an answer to the question “Do you still have a slot available”	No - Email does not contain an answer to the question “Do you still have a slot available”
Wir können Ihnen einen Platz anbieten	Wir nehmen Kinder immer zum neuen Kindergartenjahr auf
Wir haben noch ein bis zwei freie Plätze	Die Krippenplätze werden zentral vergeben
Bei uns sind noch Plätze frei	Wir nehmen Kinder ab drei Jahren auf.
Wir können Ihnen für Januar keine Zusage machen	Bitte wenden Sie sich an ...
Gruppen sind bereits voll	Wir betreuen in der Krippe nur Kinder im Alter von 1-3 Jahren
Wir haben alle Plätze besetzt	Sie stehen jetzt auf unserer Warteliste
So junge Kinder nehmen nur Kinderkrippen auf	Bitte rufen Sie doch einfach mal an
Wir haben noch Kapazitäten	Anmeldephase läuft noch.
Ich kann derzeit nicht sagen, ob ein Platz frei ist	Wohnen Sie denn in ...?
Nur noch wenige Plätze frei	Wir nehmen Kinder erst ab ...
Platzvergabe für 21/22 abgeschlossen	Ich würde ihr Kind bei uns auf die Warteliste setzen.
Warteliste jetzt schon überfüllt	Gerne nehme ich Ihr Kind auf unsere Interessentenliste auf
Für Januar noch nicht zusage.	Anmeldeverfahren beendet
Wenige Plätze verfügbar	beendet
Verfügbarkeit kommt darauf an ob ...	Für die Vergabe nicht zuständig.
die Platzvergabe wird vom Jugendamt koordiniert	
unterbringen	
Bei uns aufnehmen	
Noch verfügbar	
Noch nicht absehbar	
Platz frei	
Keine Neuaufnahmen mehr	
Nur auf die Warteliste	
Wir sind leider schon voll ausgelastet	
für einen Platz in der Krippe, müssen Sie sich	
bitte an das Rathaus wenden.	
Mit dem Stand von heute kann ich Ihnen leider	
keinen Platz für Januar 2022 anbieten.	
Wir können nur Kinder aus dem Wohnumfeld aufnehmen,	
wenn Bedarf der Mitarbeiter gedeckt ist	

Notes: Table shows the features for outcome classes “Yes” and “No” of the outcome *Slot Offer*. Features aim to provide context for the classification of the algorithm while simultaneously complying with class definitions. Out of the 12’547 observations, 798 (6.4%) were humanly labeled as slot offers and 11’749 (93.6%) as not a slot offer.

TABLE III: Features for Single Feature Baseline Comparison

Category	Yes class label	No class label
Waiting List - Single Label Combination 1	Wir können Sie auf die Warteliste setzen	Die Aufnahme ist nicht möglich
Waiting List - Single Label Combination 2	Wir können Ihnen einen Platz auf der Warteliste anbieten	Eine Aufnahme ist derzeit nicht möglich
Answer - Single Label Combination 1	Wir haben noch einen Platz frei	Wir haben keinen Platz frei
Answer - Single Label Combination 2	Wir haben einen Platz anzubieten	Wir haben keinen Platz mehr anzubieten

Notes: Table shows the features for outcome classes “Yes” and “No” of the outcome *Answer 2*. Features may or may not be citations from actual emails (frequently used expressions and key words) and aim to provide context for the classification of the algorithm while simultaneously complying with class definitions. Out of the 12’547 observations, 10’148 (80.9%) were humanly labeled as containing an answer to the question “How can we apply for a slot?” and 2’399 (19.1%) as not containing an answer to the question.

APPENDIX B

APPENDIX – PSEUDO CODE FOR MAIN ALGORITHM AND BASELINES

A. Pseudo Code Main Algorithm

Algorithm 1 Pseudo Code - Main Algorithm in Figure 1

Require:

text_instances – Text instances
features_cat1 – Feature list of class 1
features_cat2 – Feature list of class 2
agg – Calculate the feature mean and aggregate it
certainty_lvl – Certainty level
confidence_lvl – Prediction Confidence
ord – Feature ordering and permutation method

Ensure: Classified class or "undecided"

```

DECLARE agg_prob_1 = 0
DECLARE agg_prob_2 = 0
DECLARE prob_reduction = 0.5
                                ▷ Order the feature combinations
DECLARE perms = ord(features_cat1, features_cat2)
                                ▷ Begin classification
                                ▷ Repeat for each text instance
for instances i in text_instances do
                                ▷ Repeat for each feature combination
    for perm p in perms do
        ▷ Calculate the probability score of each feature

        prob_1, prob_2 = calculate_probs(text_instancei, p)
        ▷ Checks if difference between class features is larger Prediction
        Confidence
        if abs(prob_1 - prob_2) > confidence_lvl then

            prob_1 = prob_1 - prob_reduction
            prob_2 = prob_2 - prob_reduction
                                ▷ Aggregate feature means

            agg_prob_1 = agg(agg_prob_1, prob_1)
            agg_prob_2 = agg(agg_prob_2, prob_2)
                                ▷ Make classification decision
                                ▷ Check if classification criteria is met
            if agg_prob_1 > certainty_lvl then
                return Class 1
            else if agg_prob_2 > certainty_lvl then
                return Class 2
            end if
        end if
    end for
end for
                                ▷ For all unclassified text instances
return Undecided

```

B. Pseudo Code Baselines

Algorithm 2 Pseudo Code - Top n Features Baseline described in Section V-A0b

Require:

text_instances – Text instances
features_class1 – Feature list of class 1
features_class2 – Feature list of class 2
feature_lists – Combined feature lists of class 1 and 2

Ensure: Classified category

```

sort(predictions_each_feature)
DECLARE top_n =
min(length(features_cat1), leng(features_cat2))
if top_n is even then
    top_n = top_n + 1
end if
                                ▷ Begin classification
                                ▷ Repeat for each text instance
for instances i in text_instances do
    ▷ Calculate the probability score of combined feature lists
    prob_1, prob_2 = calculate_probs(text_instancei, feature_lists)
    ▷ Take the n number of features with highest probability
    order_predictions = ord(predictions_each_feature)
    top_n_list = order_predictions[0..top_n - 1]
    ▷ Calculate the probability score of "Top n" features
    prob_1, prob_2 = calculate_probs(text_instancei, top_n_list)
    ▷ Implement majority voting
    count_cat1, count_cat2 =
    count_by_cat(top_n_list)
                                ▷ Make classification decision
    if count_cat1 > count_cat2 then
        return 1
    else
        return 2
    end if
end for

```

Algorithm 3 Pseudo Code - Decisive Feature Baseline described in Section V-A0c

Require:

text_instances – Text instances
features_class1 – Feature list of class 1
features_class2 – Feature list of class 2

Ensure: Classified category

DECLARE *permutations* = [*features_class1*] * [*features_class2*]
DECLARE *max_prob_perm*

▷ Begin classification
 ▷ Repeat for each text instance

```

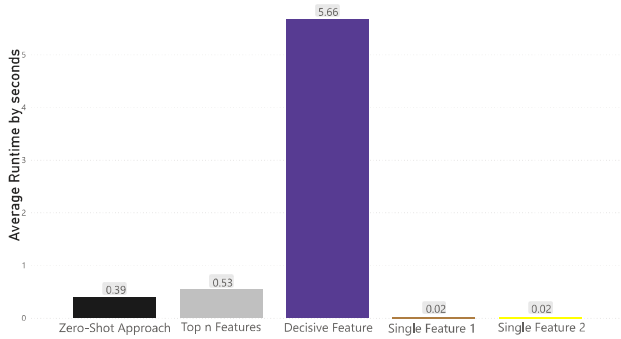
for instances i in text_instances do
  ▷ Find most decisive label combination
  for perm p in permutations do
    ▷ Make class prediction
    curr_perm_prob = calculate_prob(text_seq, p)
    ▷ Choose the most decisive prediction
    if curr_perm_prob > max_prob_perm then
      max_prob_perm = curr_perm_prob
    end if
  end for
  ▷ Make classification decision
  if max_prob_cat(max_prob_perm) = cat1 then
    return 1
  else
    return 2
  end if
end for

```

APPENDIX C

APPENDIX – RUNTIME OF BASELINES AND THE APPROACH

Fig. 4: Runtime of Baselines and the Approach



Note: Figure shows average runtimes in seconds for our machine configurations for each approach per text instance. The runtime for our Zero-Shot approach is calculated for a *Certainty Level* of 2 and a *Prediction Confidence* of 0.8.