

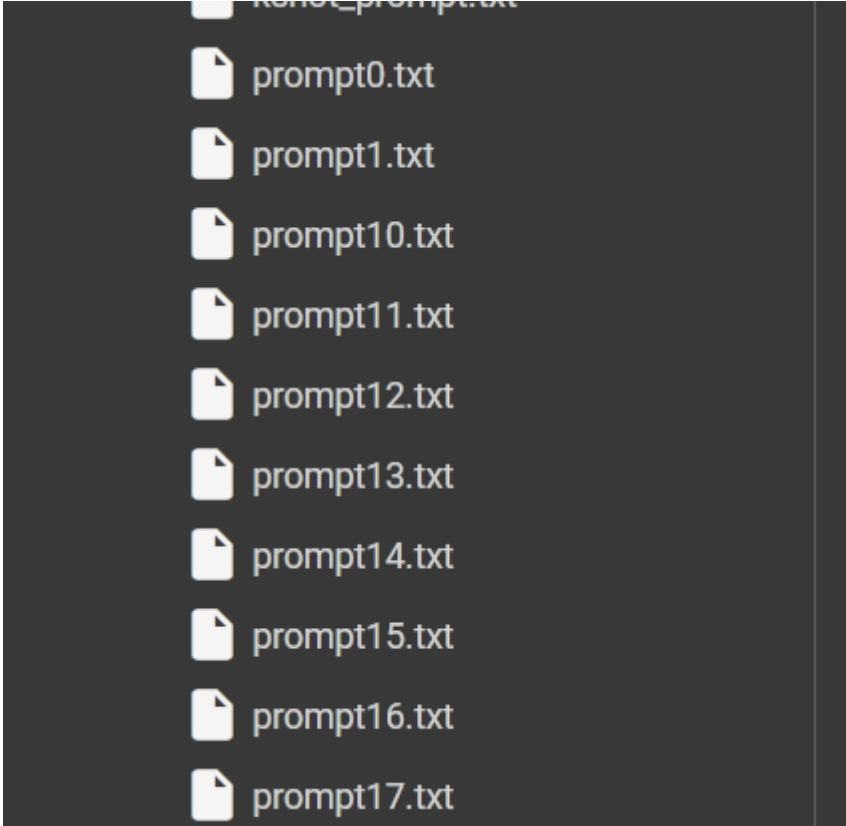
Symbol Tuning Benchmark

Symbol tuning

- Change labels to numbers.
- Trying to not mention anything related to 'important' or 'not important' tags in the prompt.

Symbol tuning

- Cleaned the code.
- Saving each prompt for more analyses.



A screenshot of a file explorer window with a dark background. It shows a list of files in a directory. The files are: `remove_prompt.txt`, `prompt0.txt`, `prompt1.txt`, `prompt10.txt`, `prompt11.txt`, `prompt12.txt`, `prompt13.txt`, `prompt14.txt`, `prompt15.txt`, `prompt16.txt`, and `prompt17.txt`. Each file is preceded by a small white icon representing a text file.

Symbol tuning

- *First step:*
 - Change the labels '1's to '58's.
 - Change the labels '0's to '47's.
- We tried to use labels that the model hasn't seen.
- So, it doesn't use its' predefined knowledge to tag news with 'important' or 'not important' tags.

Symbol tuning

- Surprisingly the Aya LLM tends to generate '58' more than '47' ones.
- This might be because there is more details or definition defined about '58' label.
- Or this might be caused by '58' being the first label.
- Or the dataset being imbalanced!
- The result shown is in 'k=20' mode.

```
test_df_counter is 24
answer of row 24 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 25
...
answer of row 25 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 26
answer of row 26 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 27
answer of row 27 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 28
answer of row 28 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 29
answer of row 29 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 30
answer of row 30 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
dataframe saved to csv file at iteration 30
test_df_counter is 31
answer of row 31 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 32
answer of row 32 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 33
answer of row 33 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 34
answer of row 34 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 35
answer of row 35 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 36
answer of row 36 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 37
answer of row 37 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 38
answer of row 38 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 39
answer of row 39 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 40
answer of row 40 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
dataframe saved to csv file at iteration 40
test_df_counter is 41
answer of row 41 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 42
answer of row 42 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 43
answer of row 43 is 47 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 44
answer of row 44 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 45
answer of row 45 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
test_df_counter is 46
answer of row 46 is 58 and k is 20.      Text type: only_title  Real tag: 1.0
test_df_counter is 47
answer of row 47 is 58 and k is 20.      Text type: only_title  Real tag: 0.0
```

Symbol tuning

- The result shown here is with k=0 shot prompts.
- The model only generates '58' as an answer!
- We can interpret two things from the observation:
 - First the k shot example help the model to obtain knowledge about '47' labels therefore resulting to predict some titles as 'not important' or '47'.
 - Second, we should include in prompt what is 'not important' or '47' label, only including information about what is known as 'important' result in generating only 'important' labels.

```
46 print(f"dataframe saved to csv file at iteration {i}")

... test_df_counter is 0
    answer of row 0 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    dataframe saved to csv file at iteration 0
    test_df_counter is 1
    answer of row 1 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    test_df_counter is 2
    answer of row 2 is 58 and k is 0.      Text type: only_title  Real tag: 1.0
    test_df_counter is 3
    answer of row 3 is 58 and k is 0.      Text type: only_title  Real tag: 1.0
    test_df_counter is 4
    answer of row 4 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    test_df_counter is 5
    answer of row 5 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    test_df_counter is 6
    answer of row 6 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    test_df_counter is 7
    answer of row 7 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    test_df_counter is 8
    answer of row 8 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    test_df_counter is 9
    answer of row 9 is 58 and k is 0.      Text type: only_title  Real tag: 0.0
    test_df_counter is 10
    answer of row 10 is 58 and k is 0.     Text type: only_title  Real tag: 1.0
    dataframe saved to csv file at iteration 10
    test_df_counter is 11
    answer of row 11 is 58 and k is 0.     Text type: only_title  Real tag: 0.0
    test_df_counter is 12
    answer of row 12 is 58 and k is 0.     Text type: only_title  Real tag: 0.0
    test_df_counter is 13
    answer of row 13 is 58 and k is 0.     Text type: only_title  Real tag: 0.0
    test_df_counter is 14
    answer of row 14 is 58 and k is 0.     Text type: only_title  Real tag: 0.0
    test_df_counter is 15
```

Symbol tuning

- The result shown here is with k=1 shot prompts.
- The model generates '47' labels sporadically.
- This means that one example provided in the prompt was not enough to give the model enough information to predict more labels as '47'.
- But it shows that even providing one example can change the output!

```
answer of row 0 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
dataframe saved to csv file at iteration 0
test_df_counter is 1
answer of row 1 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 2
answer of row 2 is 58 and k is 1.      Text type: only_title  Real tag: 1.0
test_df_counter is 3
answer of row 3 is 58 and k is 1.      Text type: only_title  Real tag: 1.0
test_df_counter is 4
answer of row 4 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 5
answer of row 5 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 6
answer of row 6 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 7
answer of row 7 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 8
answer of row 8 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 9
answer of row 9 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 10
answer of row 10 is 47 and k is 1.      Text type: only_title  Real tag: 1.0
dataframe saved to csv file at iteration 10
test_df_counter is 11
answer of row 11 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 12
answer of row 12 is 58 and k is 1.      Text type: only_title  Real tag: 0.0
test_df_counter is 13
```

Symbol tuning

- The result shown here is with k=50 shot prompts.
- The model generates more '47' labels.
- The results shows that the information and details about the 'not important' news is a necessity to override LLM predefined knowledge.

```
test_df_counter is 19
answer of row 19 is 47 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 20
answer of row 20 is 47 and k is 50.      Text type: only_title  Real tag: 0.0
dataframe saved to csv file at iteration 20
test_df_counter is 21
answer of row 21 is 58 and k is 50.      Text type: only_title  Real tag: 1.0
test_df_counter is 22
answer of row 22 is 47 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 23
answer of row 23 is 58 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 24
answer of row 24 is 47 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 25
answer of row 25 is 58 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 26
answer of row 26 is 58 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 27
answer of row 27 is 58 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 28
answer of row 28 is 58 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 29
answer of row 29 is 47 and k is 50.      Text type: only_title  Real tag: 0.0
test_df_counter is 30
answer of row 30 is 58 and k is 50.      Text type: only_title  Real tag: 0.0
dataframe saved to csv file at iteration 30
```


Symbol tuning

- The challenge to make predictions more accurate is to include clear definition and details for both 'important' and 'not important' news.
- This causes the language model to rely more on the information given in the prompt (or, as we know, in-context learning) rather than on its prior knowledge.

Symbol tuning results

- Results for $k = 0$ shot learning:

K = 0	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	17%	14%	93%	24%	96	5

- The shown results is for first 101 entities in test data.

Symbol tuning results

- Results for $k = 1$ shot learning:

K = 1	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	48%	19%	86%	31%	63	38

- The shown results is for first 101 entities in test data.

Symbol tuning results

- Results for $k = 5$ shot learning:

K = 5	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	47%	16%	64%	25%	58	43

- The shown results is for first 101 entities in test data.

Symbol tuning results

- Results for $k = 20$ shot learning:

K = 20	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	49%	15%	57%	24%	54	47

- The shown results is for first 101 entities in test data.

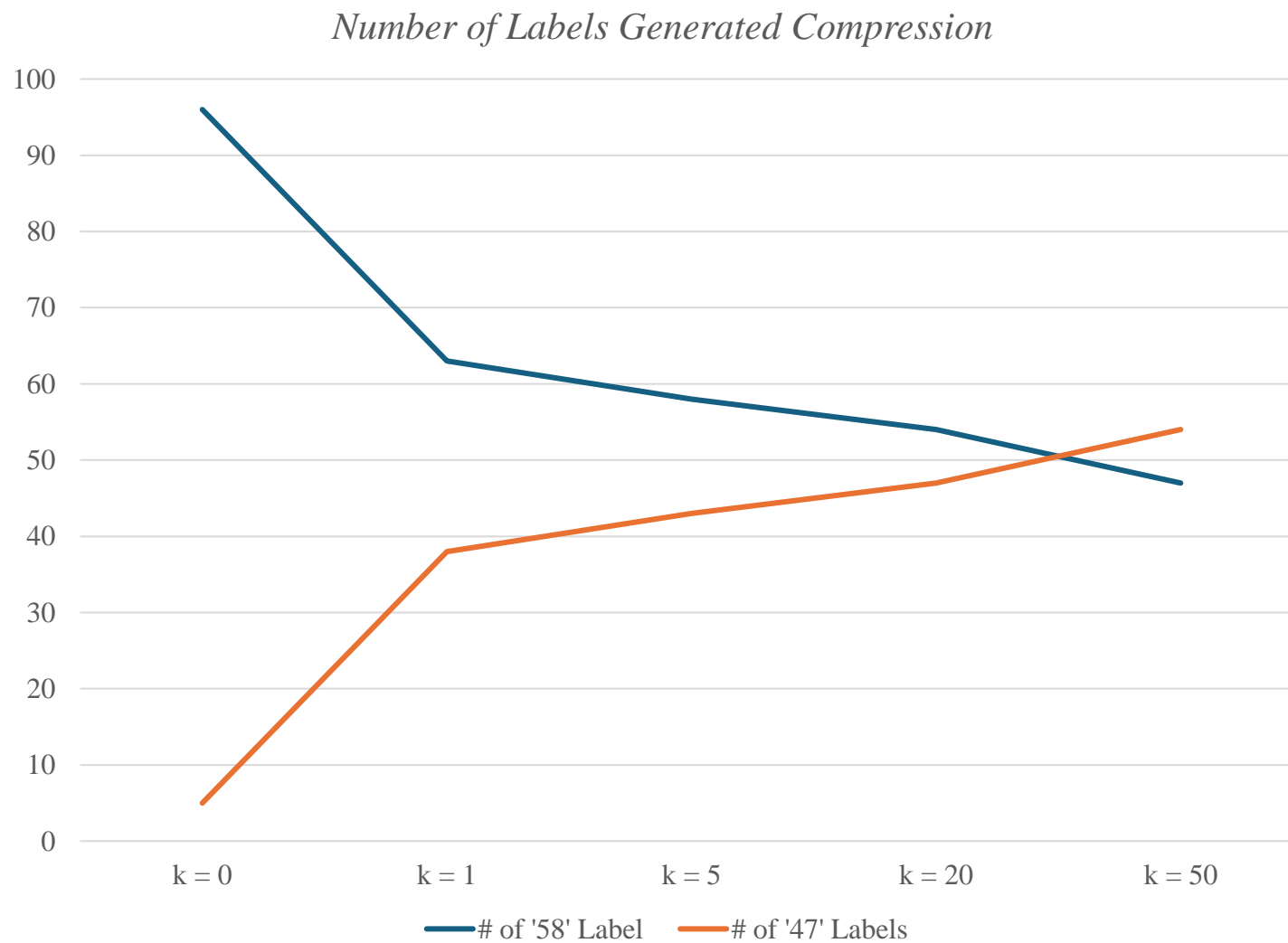
Symbol tuning results

- Results for k = 50 shot learning:

K = 50	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	55%	17%	57%	26%	47	54

- The shown results is for first 101 entities in test data.

Symbol tuning results



Symbol tuning feasible improvements

- Possible future improvements:
 - Change the prompt: The problem observed here is that the prompt lacks a definition for 'important' news but details and definitions for 'not important' ones.
 - Changing the 'important' label to something that is harder to generate because our dataset is imbalanced, and we have little 'important' news compared to 'not important' ones. Therefore, it is logical to make the 'important' label harder to generate for the LLM model.
 - Including in the prompt that we have way less 'important' news than 'not important' ones; therefore, the model should be more sensitive and conservative in generating the 'important' label.
 - Including the chain of thoughts context with the examples provided in the prompt to make the decision for the model more logical and with more reasoning information.

Symbol tuning results

- Here, we analyze the results achieved from the first 400 indices from our test dataset.
 - The number of total '1' labels is 77, and for '0' labels is 323.

# of '1' labels	# of '0' labels
77	323

Symbol tuning results

- Results for $k = 0$ shot learning:

K = 0	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	21%	19%	97%	32%	387	13

- The shown results is for first 400 entities in test data.
- The high percentage achieved by the model in recall metrics is due to mostly predicting '58' labels.

Symbol tuning results

- Results for $k = 1$ shot learning:

K = 1	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	53%	24%	69%	36%	218	182

- The shown results is for first 400 entities in test data.

Symbol tuning results

- Results for k = 5 shot learning:

K = 5	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	49%	24%	78%	37%	249	151

- The shown results is for first 400 entities in test data.
- Highest f1-score reached!

Symbol tuning results

- Results for k = 20 shot learning:

K = 20	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	48%	23%	71%	34%	242	158

- The shown results is for first 400 entities in test data.

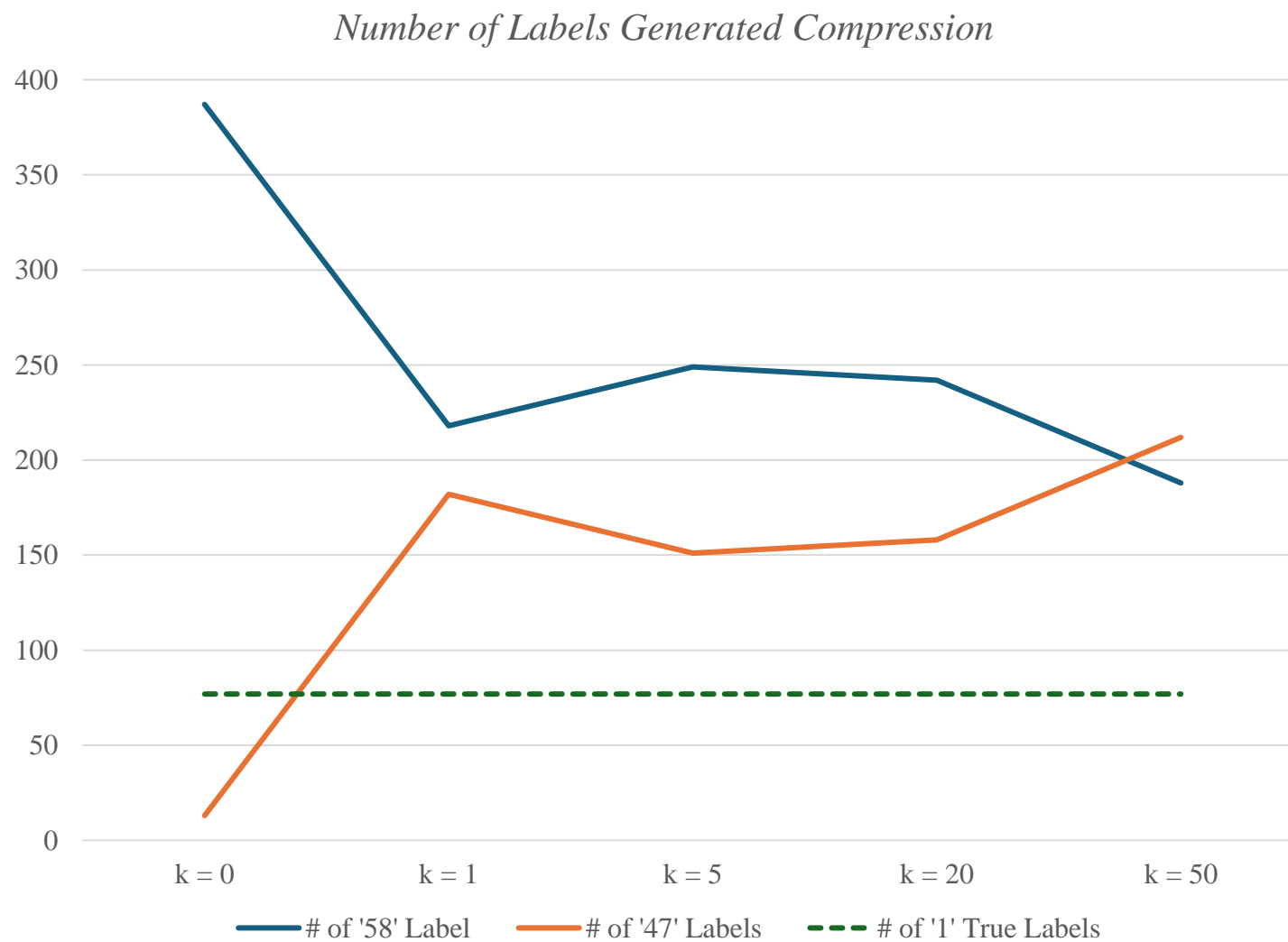
Symbol tuning results

- Results for k = 50 shot learning:

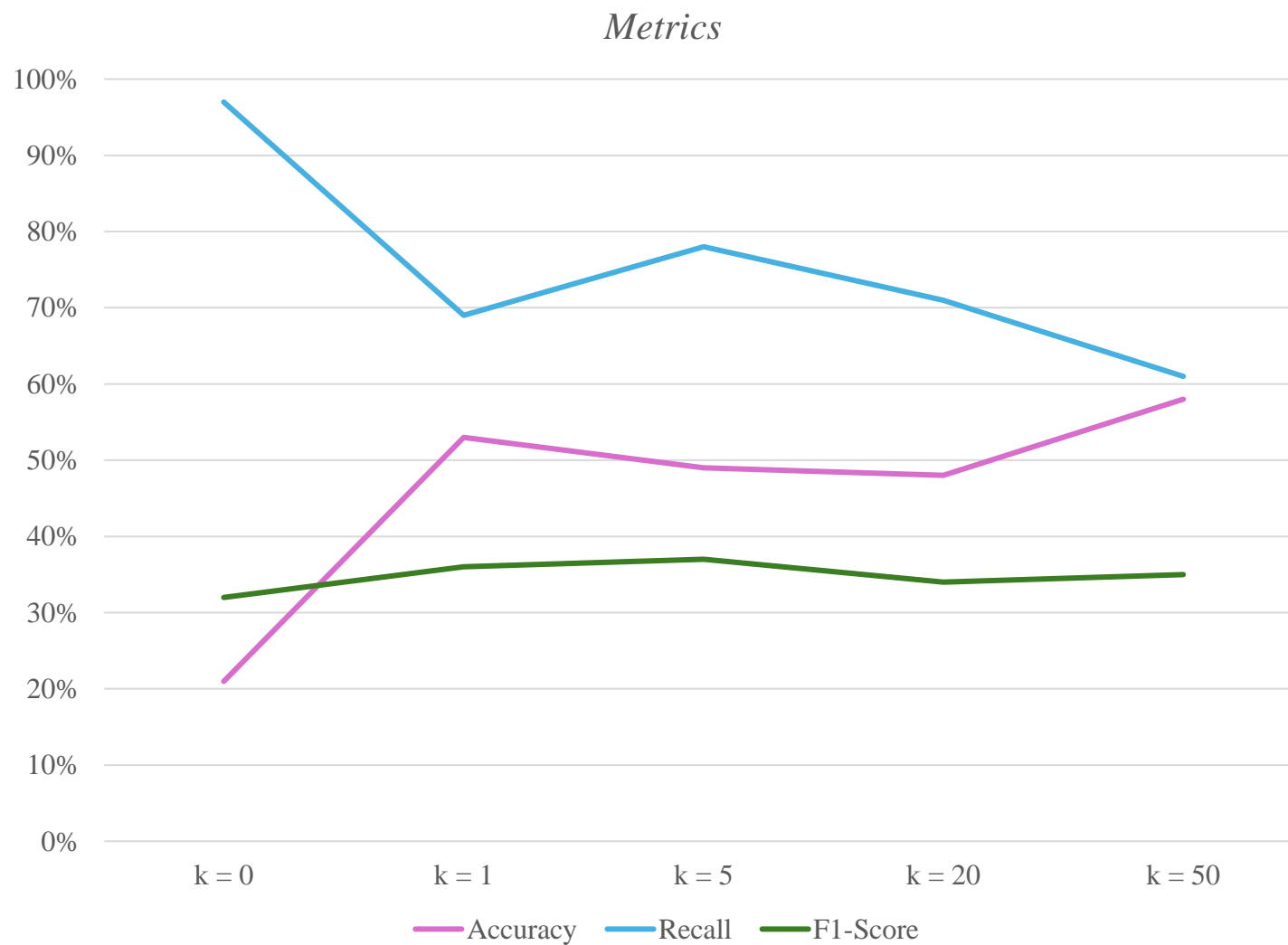
K = 50	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
Title	57%	25%	61%	35%	188	212

- The shown results is for first 400 entities in test data.
- Highest number of '47' predicted!

Symbol tuning results



Symbol tuning results



Symbol tuning results

- Now, we analyze the whole data in test dataset.
 - The total number of labeled news is '1179'
 - The number of total '1' labels is 196, and for '0' labels is 983.

# of '1' labels	# of '0' labels
196	983

Symbol tuning results

- Results for $k = 0$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 0$	19%	17%	97%	28%	1121	40

Symbol tuning results

- Results for $k = 1$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 1$	53%	21%	67%	32%	609	552

Symbol tuning results

- Results for $k = 5$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 5$	49%	21%	77%	33%	701	460

Symbol tuning results

- Results for $k = 20$ shot learning:

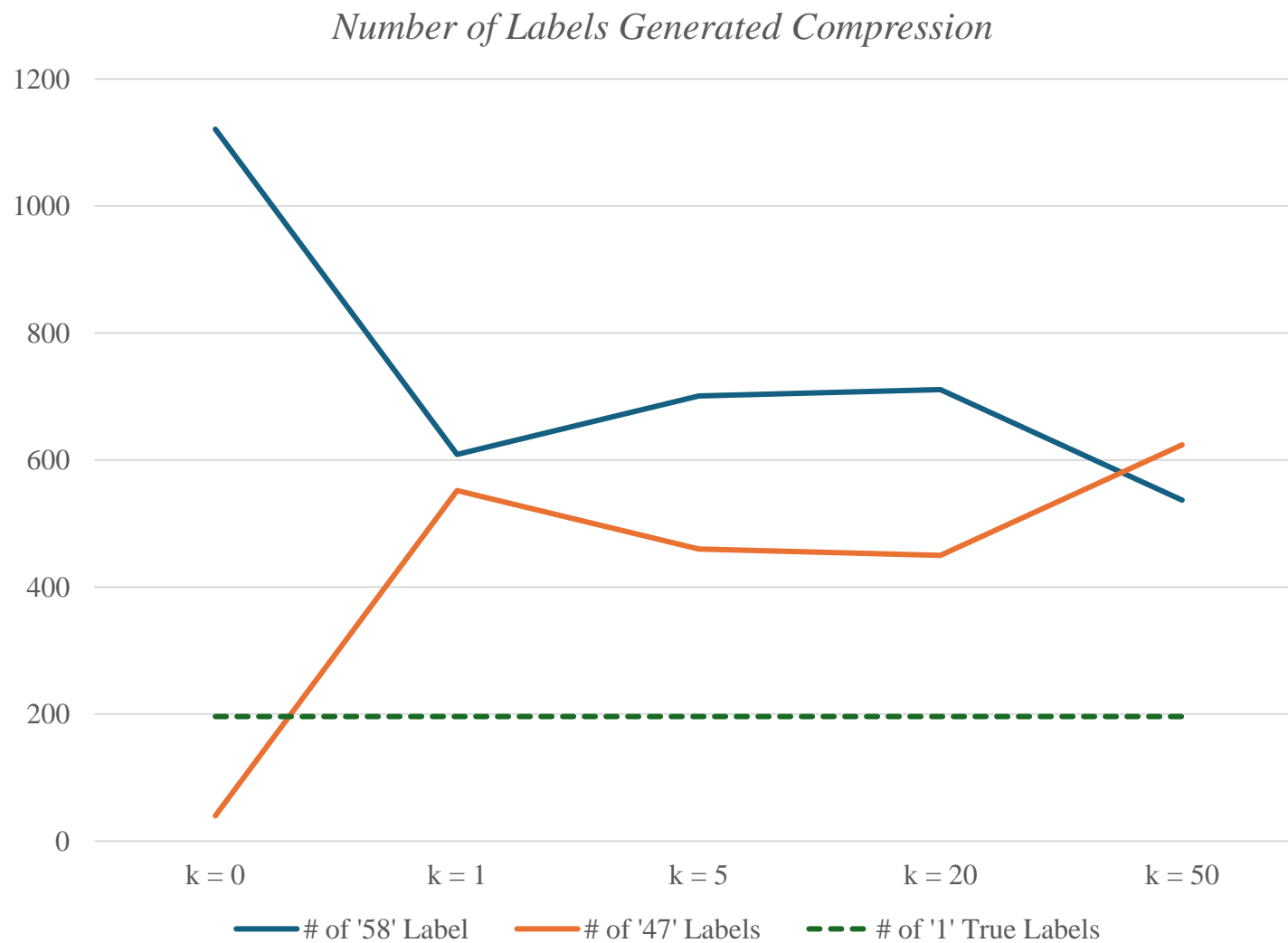
Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 20$	46%	20%	73%	31%	711	450

Symbol tuning results

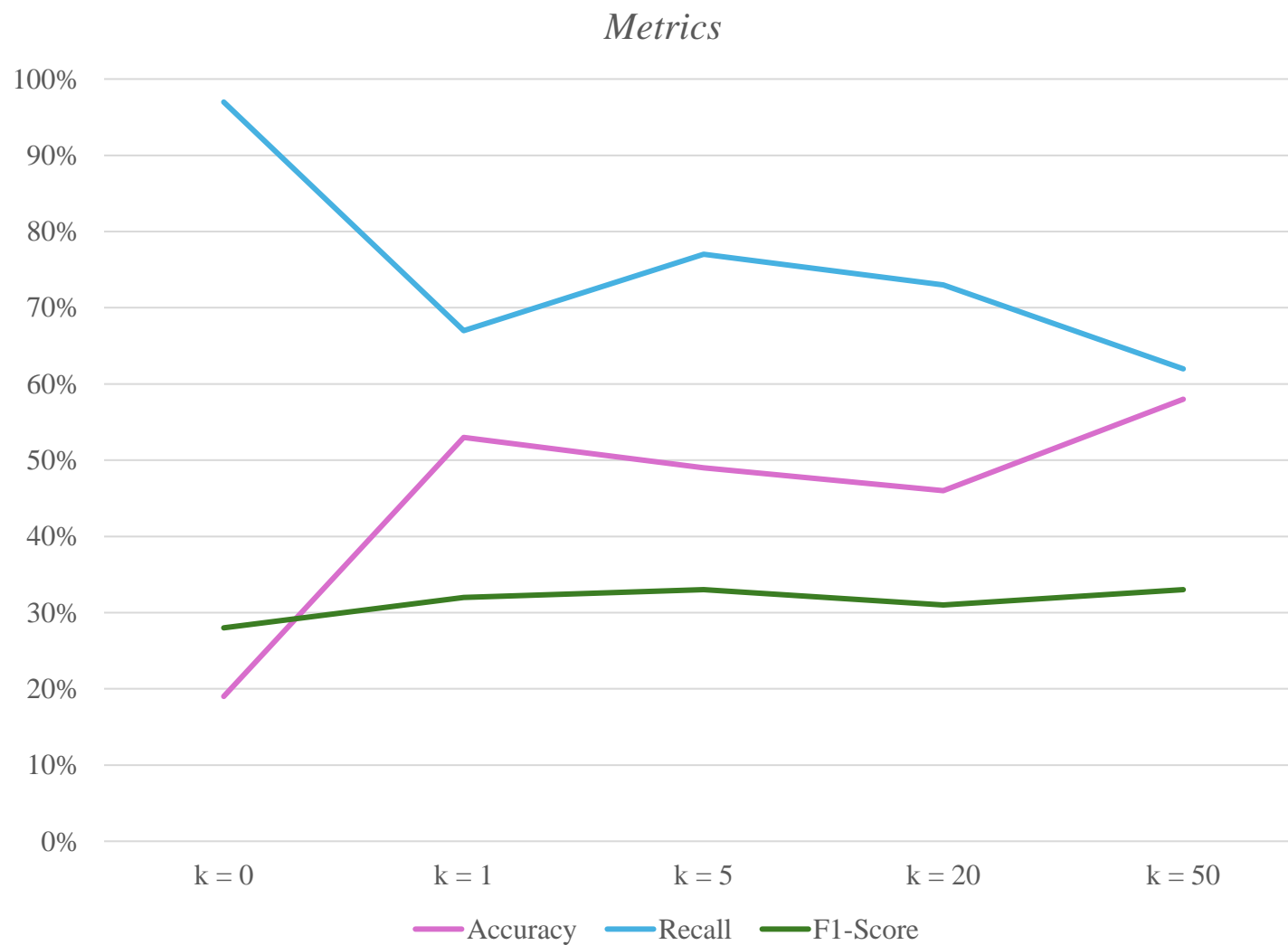
- Results for $k = 50$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 50$	58%	22%	62%	33%	537	624

Symbol tuning results



Symbol tuning results



Symbol tuning results

- All the results is shown here:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
k = 0	19%	17%	97%	28%	1121	40
k = 1	53%	21%	67%	32%	609	552
k = 5	49%	21%	77%	33%	701	460
k = 20	46%	20%	73%	31%	711	450
k = 50	58%	22%	62%	33%	537	624
Tr Labels					196	983

Symbol tuning results

- And this is the result for first 400 data in our test data:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
k = 0	21%	19%	97%	32%	387	13
k = 1	53%	24%	69%	36%	218	182
k = 5	49%	24%	78%	37%	249	151
k = 20	48%	23%	71%	34%	242	158
k = 50	57%	25%	61%	35%	188	212
Tr Labels					77	323

- This suggest that we can rely on the results achieved from first 400 samples.
 - As it can be interpreted that there is little difference between results from all samples and n=400 samples.

Symbol tuning prompt

- Now we analyze the changes made to the prompt.
- The first change is adding information and details about ‘not important’ news.
- The results can be seen in following slides.

Symbol tuning results

- As same as before we have 77 'important' labels and 324 'not important' labels in our first 401 samples from test data.

# of '1' labels	# of '0' labels
77	324

Symbol tuning results

- Results for $k = 0$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 0$	27%	20%	95%	33%	363	38

- By adding s definition for 'not important' news we observe an increase in detecting '47' labels.
- This result into mor accuracy and f1-score!

Symbol tuning results

- Results for $k = 1$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 1$	41%	23%	86%	36%	293	108

- The accuracy dropped here!
- This is because the model is less sensitive to the example provided in prompt. As we can understand the increase in number of '47' labels predicted is steadier, revealing that the model is acting more nuanced about the example provided.

Symbol tuning results

- Results for k = 5 shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
k = 5	46%	22%	71%	34%	249	152

- Here we saw a small increase to the number of '47' labels predicted.
- This means the model shows more resistance to the examples because of the change in prompt.

Symbol tuning results

- Results for $k = 20$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 20$	58%	26%	69%	38%	202	199

- Highest accuracy and f1-score achieved so far!

Symbol tuning results

- Results for $k = 50$ shot learning:

Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
$k = 50$	57%	24%	56%	33%	180	221

- With many examples provided eventually the model predicted more '47' labels than '58' ones.

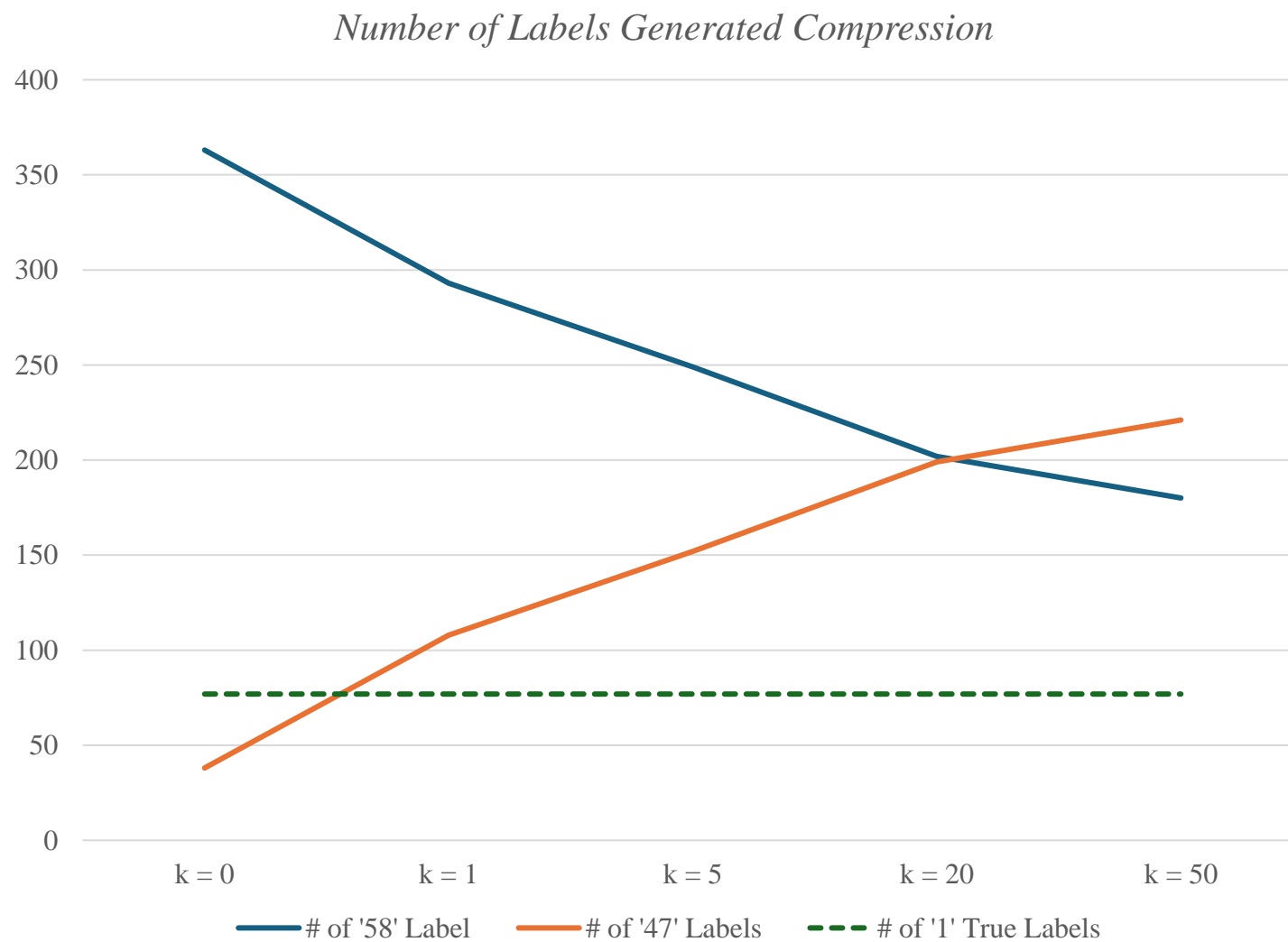
Symbol tuning results

- The whole results can be seen here:

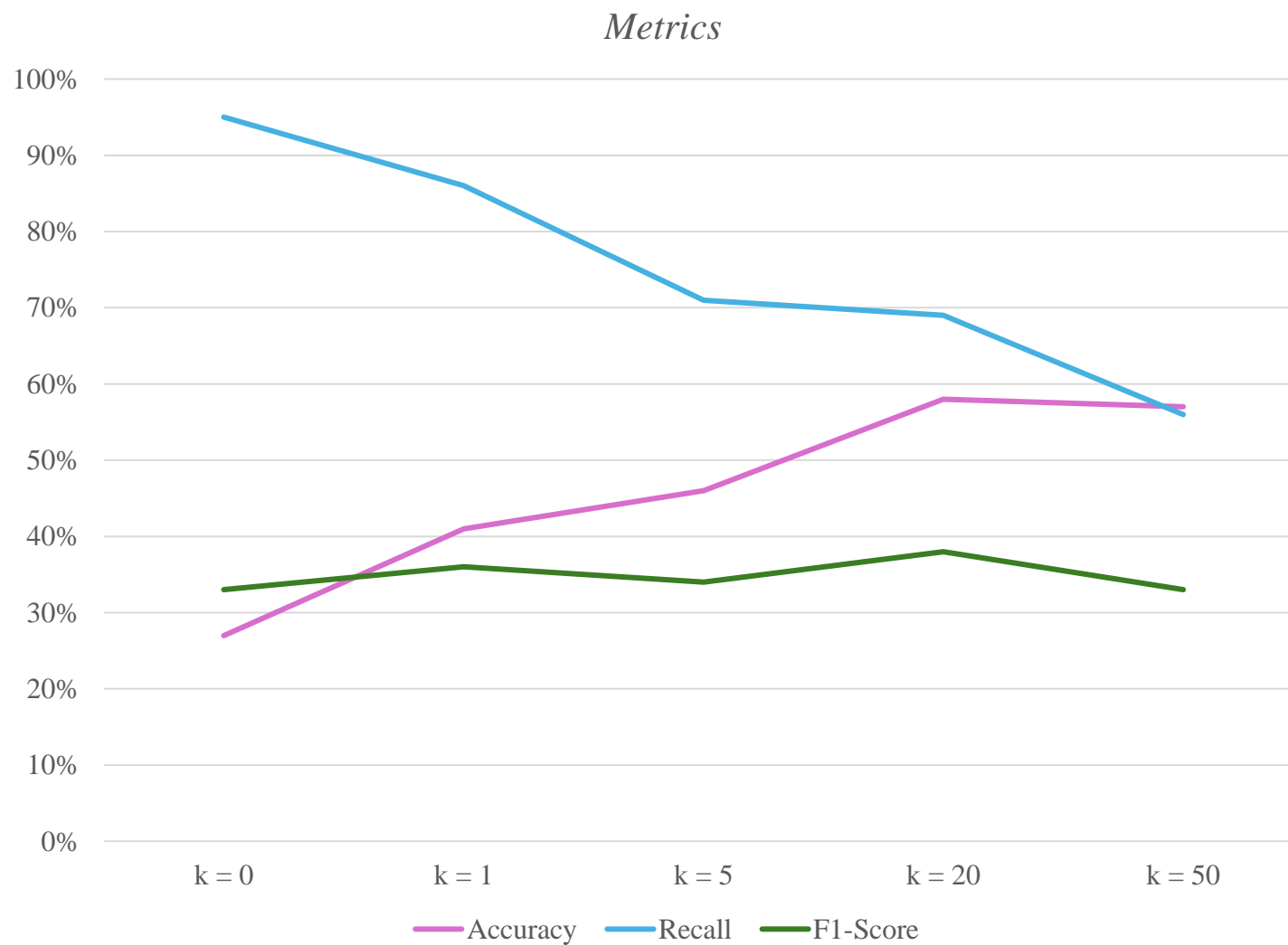
Title Only	Accuracy	Precision	Recall	F1-Score	# of '58'	# of '47'
k = 0	27%	20%	95%	33%	363	38
k = 1	41%	23%	86%	36%	293	108
k = 5	46%	22%	71%	34%	249	152
k = 20	58%	26%	69%	38%	202	199
k = 50	57%	24%	56%	33%	180	221
Tr Labels					77	323

- The accuracy and f1-score in k = 20 scenario saw the biggest increase.

Symbol tuning results



Symbol tuning results



Symbol tuning

- The results illustrate that even a small change in prompt can have dramatic alter the results and be observed in symbol tuning.