Unit 7. Natural language processing evaluation metrics

Estimated time

00:30

Overview

This unit explains how to evaluate the quality of your natural language processing (NLP) algorithm.

IBM Training

Unit objectives

- Define various metrics to measure the quality of NLP algorithms.
- Understand the difference between these metrics.

Figure 7-1. Unit objectives

System evaluation

- How can we measure the solution quality?
- Target: You developed a new search engine. You must define how well it works.

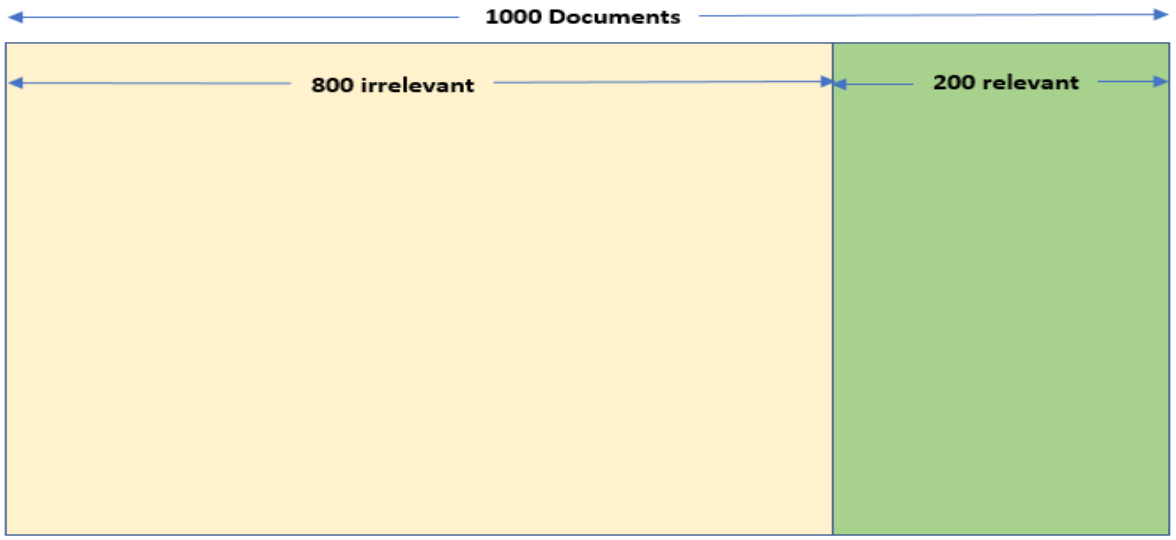

Figure 7-2. System evaluation

How can we measure the solution quality? In this presentation, we focus on a basic metric to evaluate system performance in information retrieval. Assume that you developed a search algorithm that helps you to retrieve related words from a corpus that contains 1000 documents.

From these 1000 documents, assume 200 are relevant to the word cat, and the other 800 documents are irrelevant.

System evaluation (cont.)

- You ran a search test for the word “cat”.
- After the test ran, the search engine retrieved the documents that are shown here.

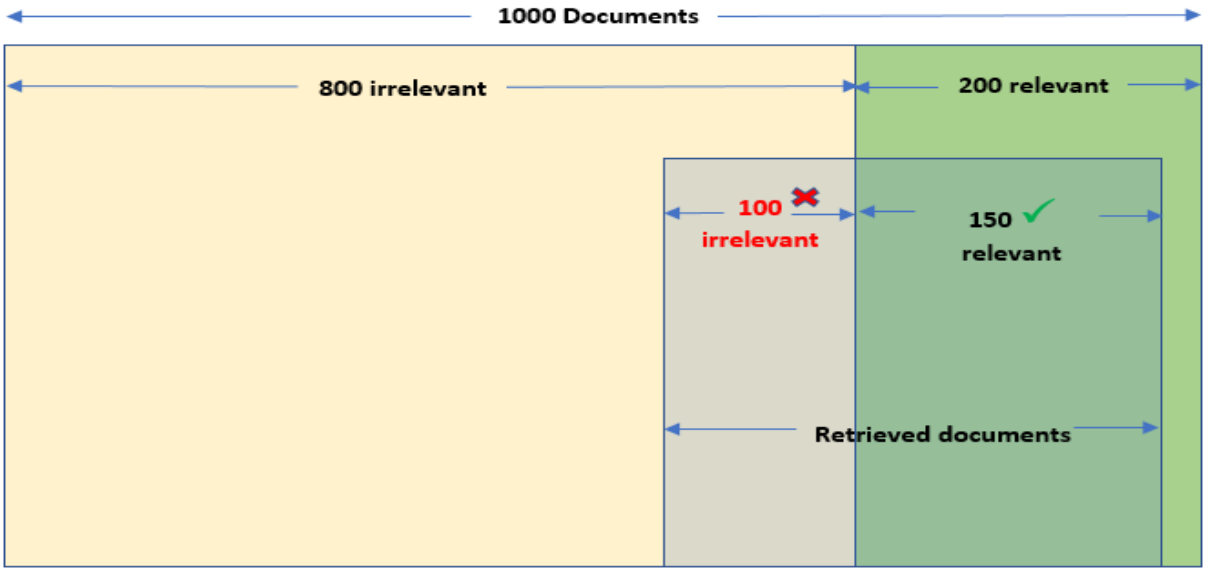

Figure 7-3. System evaluation (cont.)

You test your solution by searching for the word “cat”. Your algorithm returns 250 documents, where 150 documents are relevant (which means your algorithm missed 50 relevant documents) and 100 documents are irrelevant (which means your algorithm correctly eliminated 700 of the irrelevant documents).

IBM Training

System evaluation (cont.)

Confusion matrix

		Relevant documents in results set	Irrelevant documents in results set
Algorithm results	Retrieved	True positive (Tp)	False positive (Fp)
	Not retrieved	False negative (Fn)	True negative (Tn)

- How many **relevant** documents were **retrieved** by the algorithm?
  150 documents → True positive (Tp).
- How many **irrelevant** documents were **retrieved** by the algorithm?
  100 documents → False positive (Fp)
  (total 250 documents retrieved – 150 relevant documents).
- How many **relevant** documents did the algorithm **not retrieve**?
  50 documents → False negative (Fn).
- How many **irrelevant** documents did the algorithm **not retrieve**?
  700 documents → True negative (Tn).

Figure 7-4. System evaluation (cont.)

A confusion matrix, also known as an error matrix, is a specific table layout that enables visualization of the performance of an algorithm.

Based on the results for this example, how many **relevant** documents were **retrieved** by the algorithm? The answer is 150 documents. This value is the True positive (Tp).

Based on the results for this example, how many **irrelevant** documents were **retrieved** by the algorithm? The answer is 100 documents (total 250 documents retrieved – 150 relevant documents). This value is the False positive (Fp).

Based on the results for this example, how many **relevant** documents did the algorithm **not retrieve**? The answer is 50 documents. This value is the False negative (Fn).

Based on the results for this example, how many **irrelevant** documents did the algorithm **not retrieve**? The answer is 700 documents. This value is the True negative (Tn).

The objective is to improve the algorithm to decrease the Fp and Fn values.

IBM Training

System evaluation (cont.)

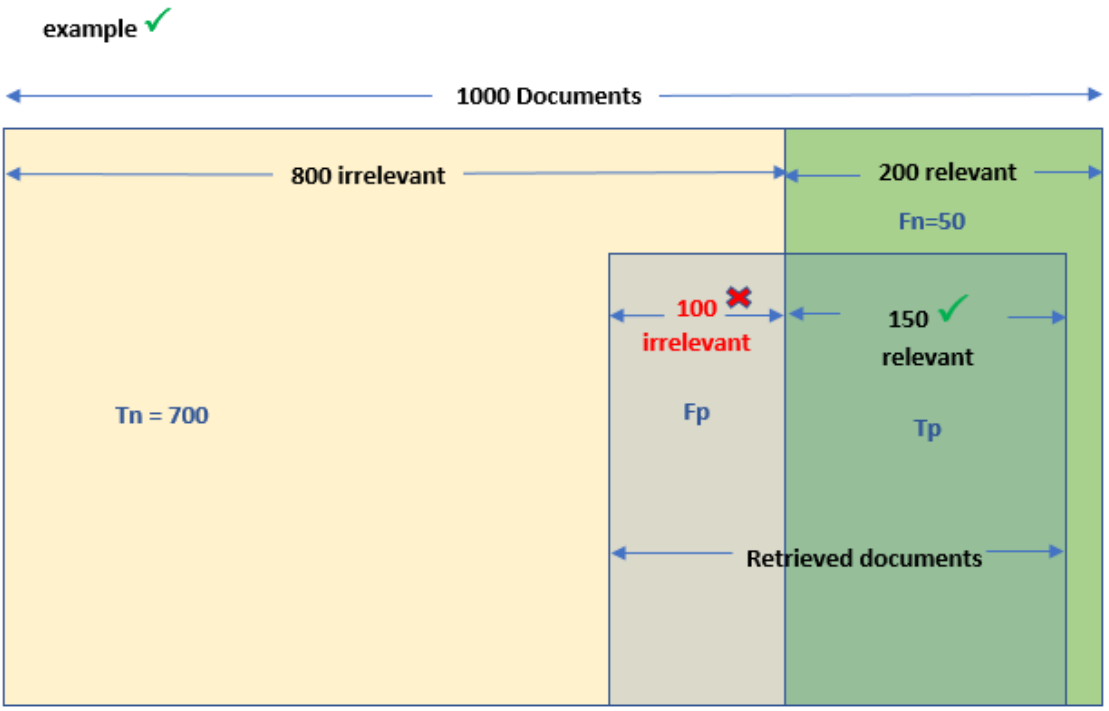

Figure 7-5. System evaluation (cont.)

We map the confusion matrix to the graph to produce the visuals for Tp, Fp, Tn, and Fn. We add Tp to the retrieved relevant documents area and Fp to the retrieved irrelevant area. We add Fn to the not retrieved relevant area and Tn to the not retrieved irrelevant area.

IBM Training

System evaluation (cont.)

Accuracy

- Calculates how many correct results your solution managed to identify.

```
Accuracy = (Tp+Tn)/(Tp+Tn+Fp+Fn)
```

- Apply the formula to the example.

```
Accuracy = (150+700)/(1000)
```

- Useful for symmetric data sets where the values of positive and negatives are almost the same.

Figure 7-6. System evaluation (cont.)

Accuracy is as a numeric measure of how good your algorithm is. It calculates how many correct results your solution managed to identify, which is the proportion of true results among the total number of cases that are examined.

Accuracy is defined by the following formula, which includes the Tp, Tn, Fp, and Fn metrics:

Accuracy = (Tp+Tn)/(Tp+Tn+Fp+Fn)

By applying the values from the example, accuracy can be calculated as follows:

Accuracy = (150+700)/1000 = 0.85

Accuracy is a good measure but only when you have symmetric data sets where the number of positive values and negatives values are almost the same. For example, if your data set is split as 90 positive samples and 10 negative samples, classifying all as positive gives a 0.90 accuracy score.

Therefore, we must look at other metrics such as precision and recall to evaluate the quality of the algorithm.

System evaluation (cont.)

Precision

- Represents the fraction of retrieved documents that are relevant.

```
Precision = Tp/(Tp+Fp)
```

- Apply the formula to the example.

```
Precision = 150/(150+100)
Precision = 150/250 = 0.60
```


Figure 7-7. System evaluation (cont.)

Precision is a numeric measure that represents the fraction of retrieved documents that are relevant. It is defined by the following formula:

Precision = Tp/(Tp+Fp)

Apply the formula to the example:

Precision = 150/(150+100)

150 retrieved relevant / 250 total retrieved = 0.60

System evaluation (cont.)

Recall

- Represents the fraction of relevant documents that were retrieved.

```
Recall = Tp/(Tp+Fn)
```

- Apply the formula to the following example.

```
Recall = (150)/(150+50)
Recall = 150/200 = 0.75
```


Figure 7-8. System evaluation (cont.)

Recall is a numeric measure that represents the fraction of relevant documents that were retrieved. It is defined by the following formula:

Recall = Tp/(Tp+Fn)

Apply the formula to the example:

Recall =150/(150+50)

150 retrieved relevant / 200 total relevant = 0.75

System evaluation (cont.)

F-Score (F-measure**)**

- Enables you to tradeoff precision against recall.
- The higher the F-score value is, the better the algorithm is.
- Here is the formula.

```
F = 2*Precision*Recall/(Precision+Recall)
```

- Apply the formula to the example.

```
F = (2*0.60*0.75)/(0.60+0.75)
F = 0.9/1.35= 0.6667
```


Figure 7-9. System evaluation (cont.)

The **F-score** (also called F-measure) is a measure that enables you to tradeoff precision against recall by approximately averaging the precision and recall values.

The formula for F-score is:

F = (2*Precision*Recall)/(Precision +Recall)

Apply the formula to the example:

F = (2*0.60*0.75)/(0.60+0.75)

= 0.9/1.35= 0.6667

IBM Training

Unit summary

- Define various metrics to measure the quality of NLP algorithms.
- Understand the difference between these metrics.

Figure 7-10. Unit summary