# TF-IDF

$$TF(x, y) = number\ of\ occurrences\ of\ keyword\ x\ within\ document\ y$$

- The more instances of our keyword we find in a document, the higher the TF value will be.

# TF-IDF

$$IDF(x) = log(\frac{total\ number\ of\ documents\ in\ the\ corpus}{number\ of\ documents\ containing\ keyword\ x})$$

- The more frequently our keyword appears across documents, the lower its IDF score becomes. This is because we want to penalize common words such as ‘a’, ‘an’, ‘is’, etc., as they tend to appear in many documents.

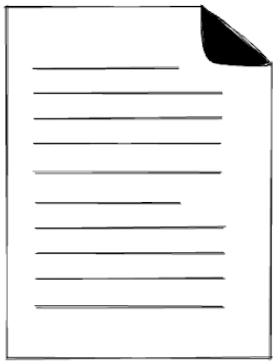
# TF-IDF

$$TF\text{–}IDF(x, y) = TF(x, y) * IDF(x)$$

- TF-IDF captures the importance of a keyword by assigning it a higher score if it appears frequently in one document but rarely in others.

# TF-IDF Issues: Keyword frequency

Document A

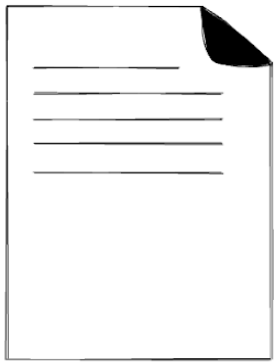


document length: 1000

keyword occurrences: 10

TF: 10

Document B



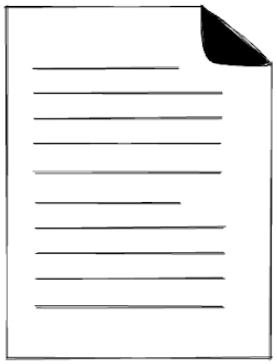
document length: 10

keyword occurrences: 1

TF: 1

# TF-IDF Issues: Keyword frequency

Document A

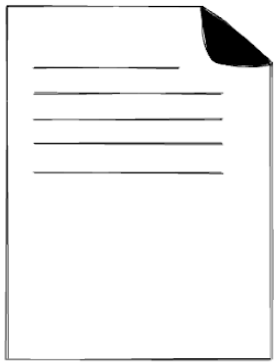


document length: 1000

keyword occurrences: 10

TF: 10

Document B



document length: 10

keyword occurrences: 1

TF: 1

$$TF(x, y) = \frac{number\ of\ occurrences\ of\ keyword\ x\ within\ document\ y}{number\ of\ words\ in\ document\ y}$$

## Documents (D)

d1 = Australia won the Cricket World Cup 2023

d2 = India and Australia played in the finals

d3 = Australia won the sixth time having last won in 2015

## Search Term

“won”

## TF

TF (“won”,d1)=1/7 = **0.14**

TF (“won”,d2)=0/7 = **0**

TF (“won, d3)= 2/10 = **0.2**

## IDF

IDF (“won”, D) = log (3/2) = **0.176**

## TF - IDF

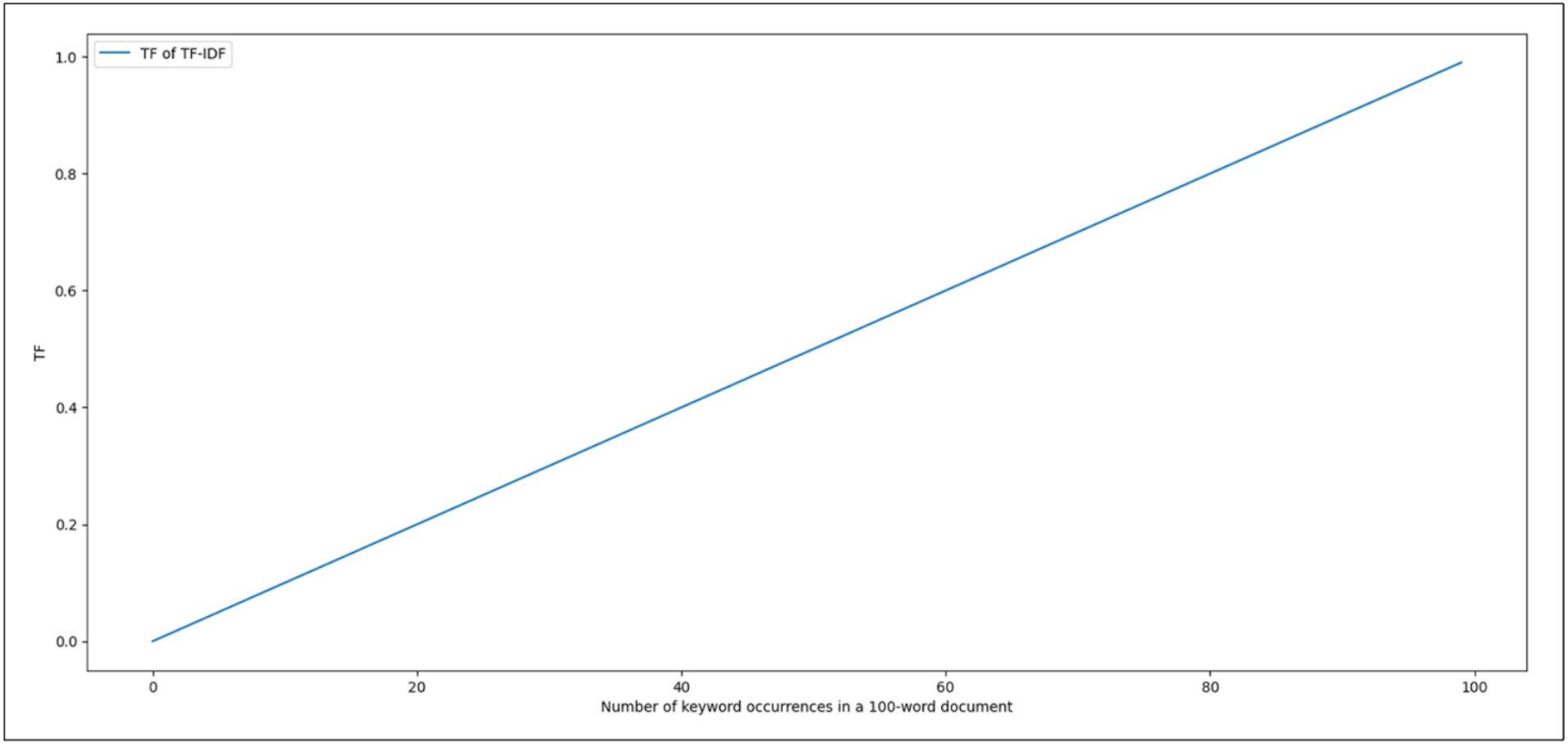
TF - IDF (“won”,d1,D)= 0.14 x 0.176 = **0.025**

TF - IDF (“won”,d2,D)= 0 x 0.176 = **0**

TF - IDF (“won, d3,D)= 0.2 x 0.176 = **0.035**

**Result - d3 > d1 > d2**

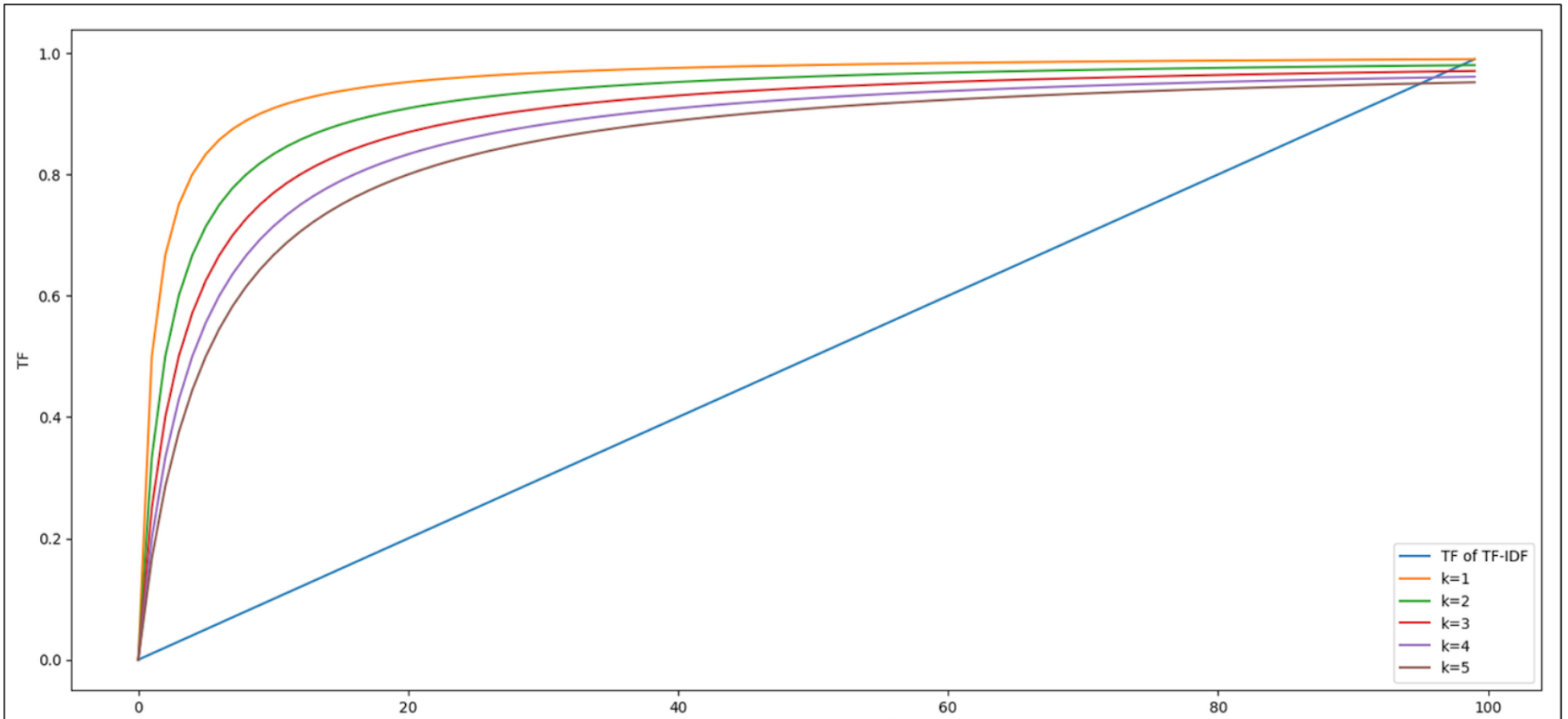
# TF-IDF Issues: Keyword Saturation




Given that we fixed the number of words in a document to 100, the TF score of TF-IDF increases linearly as the occurrence of our keyword increases.

# BM25 Improvements:: Keyword Saturation

$$TF(x,y) = \frac{TF(x,y)}{(TF(x,y)+k)}$$



# BM25 Improvements: Document Length Normalization

$$TF(x,y) = \frac{TF(x,y)}{(TF(x,y) + k * \frac{|D(y)|}{avg(D)})}$$

- The term |*D*| represents the document length, while *avg(D)* represents the average length of documents in our corpus.

- If the document is shorter than average, the value of TF/(TF+k) will increase, and vice versa. In other words, shorter documents will approach the saturation point quicker than longer documents.

# BM25 Improvements: Document Length Normalization

$$TF(x,y) = \frac{TF(x,y)}{(TF(x,y) + k * (1 - b + b * \frac{|D(y)|}{avg(D)}))}$$

If we set the value of *b* to 0, then the ratio *D/avgD* wouldn't be considered at all, meaning that we don't put any importance on the document's length. Meanwhile, the value of 1 indicates that we put a lot of importance on the document's length

# BM25 Improvements

$$IDF(x) = log(\frac{N - DF(x) + 0.5}{DF(x) + 0.5})$$