Klasifikacija dokumenata

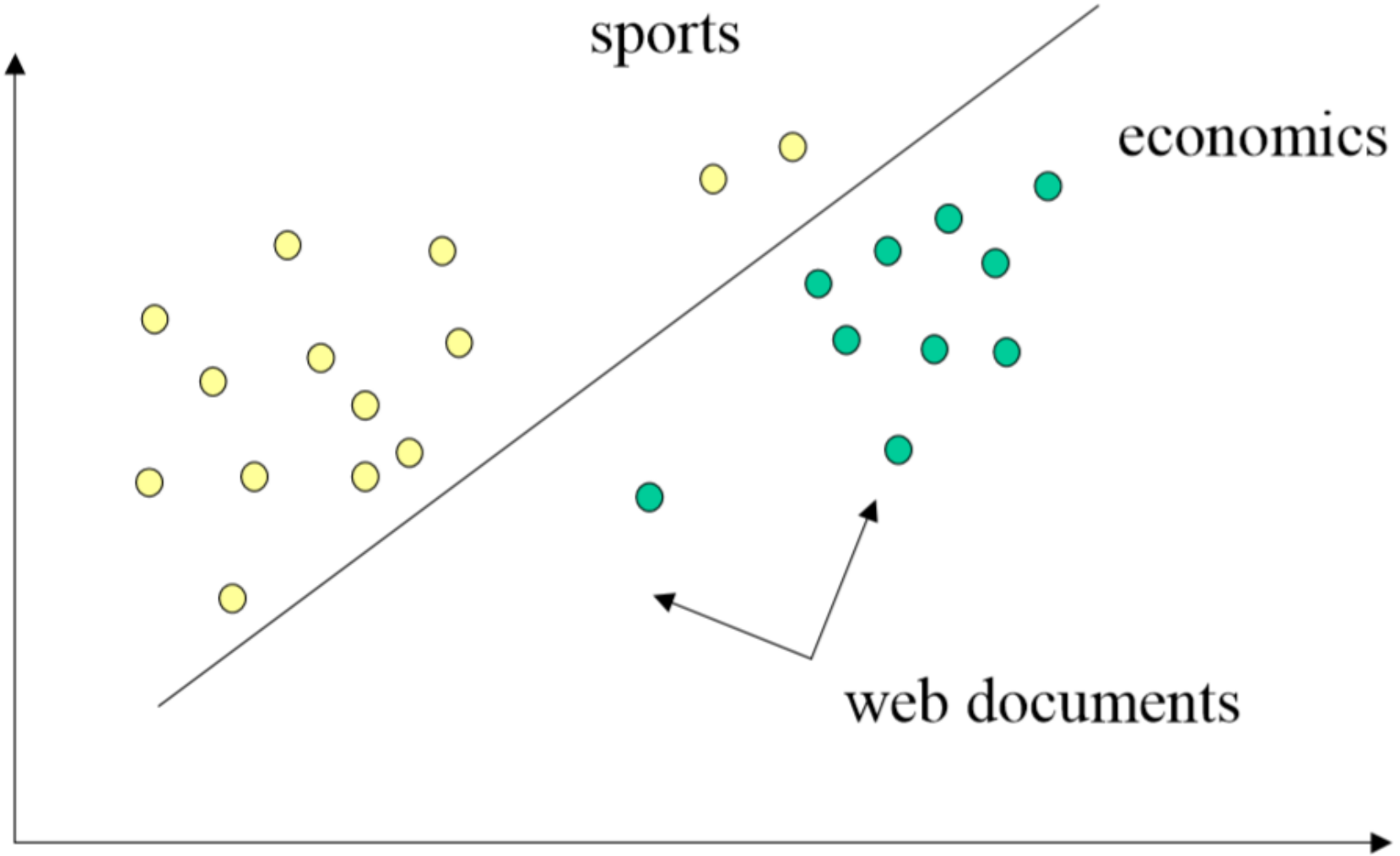
Uvod

- Text klasifikacija
  - Klasifikacija dokumenata u skup unaprijed poznatih klasa
- Alternativni nazivi
  - Text kategorizacija
  - Klasifikacija dokumenata
  - Kategorizacija dokumenata
- Dva pristupa
  - Ručna klasifikacija
  - Automatska klasifikacija

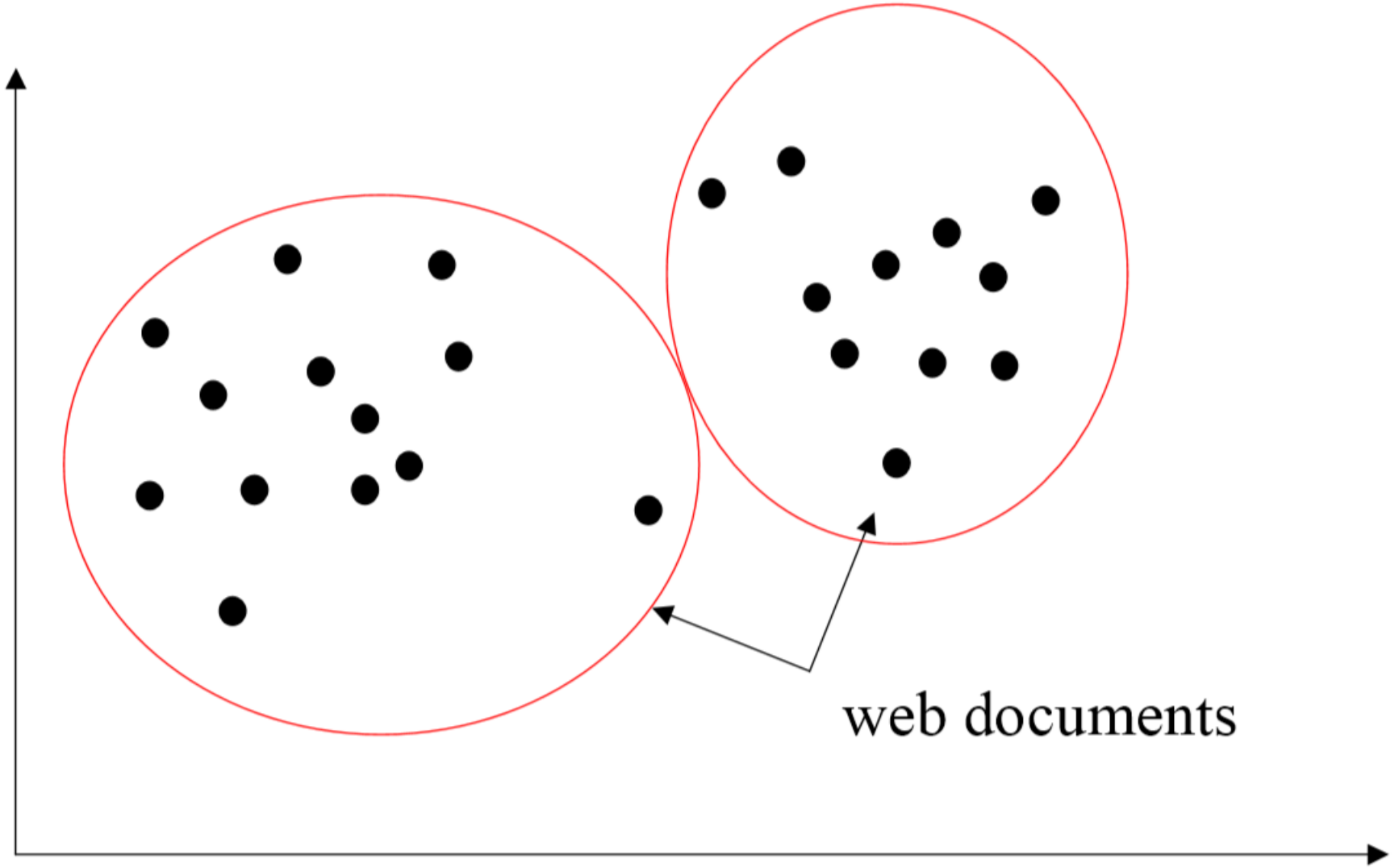
Tehnologije

- Klasifikacija
- Klasterizacija
- Information extraction
- Information retrieval
- Information filtering

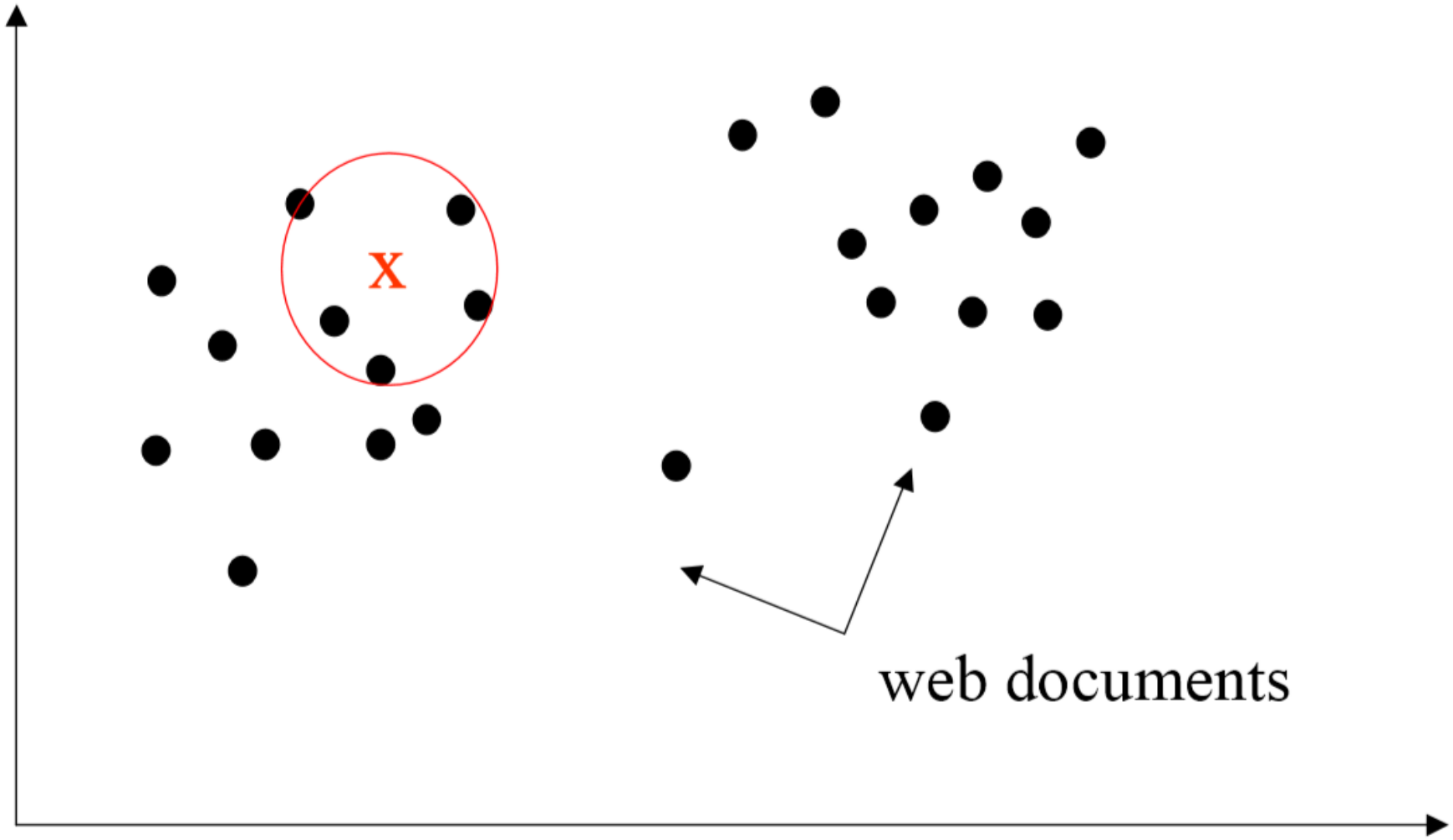
Klasifikacija



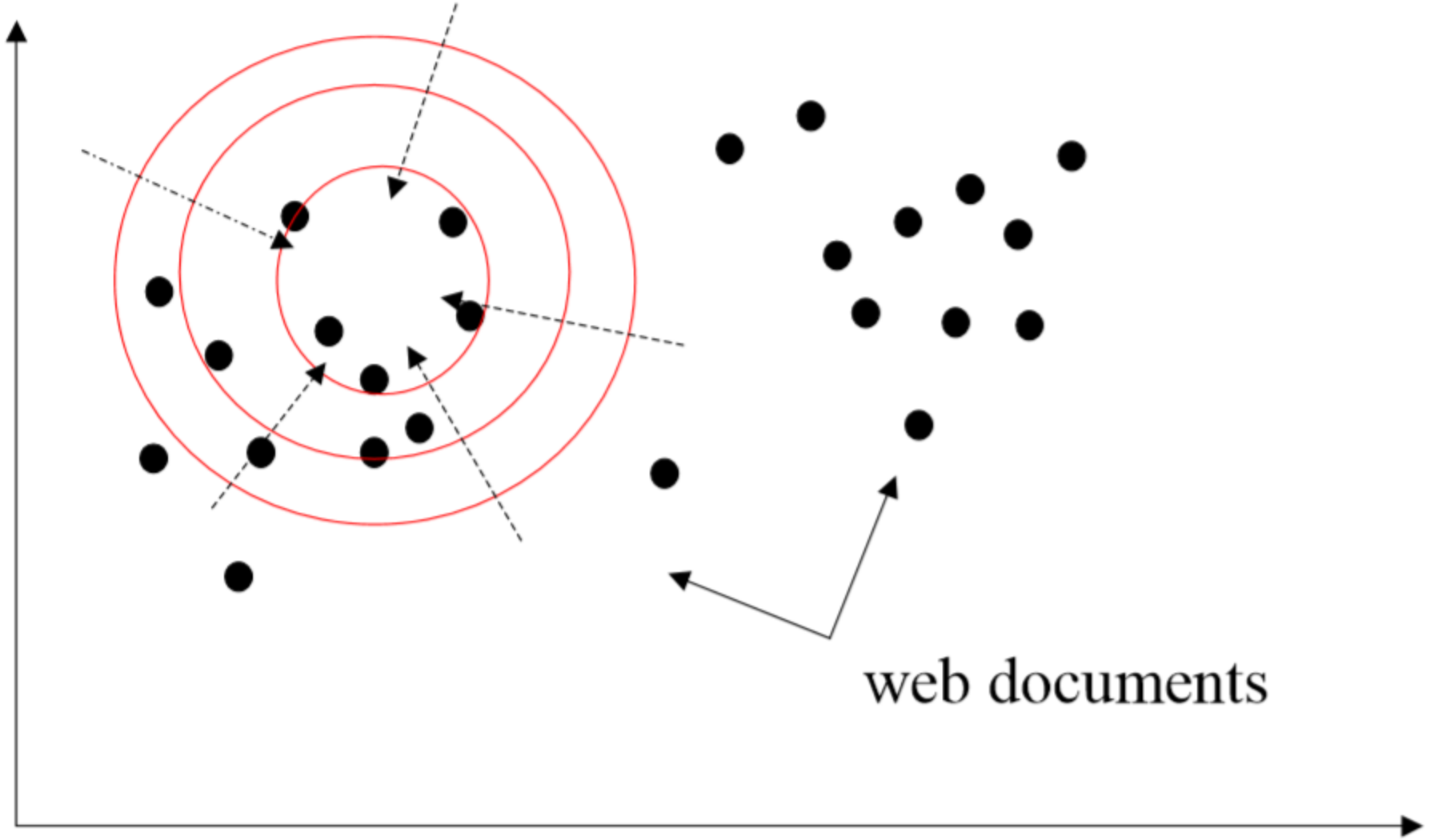
Klasterizacija



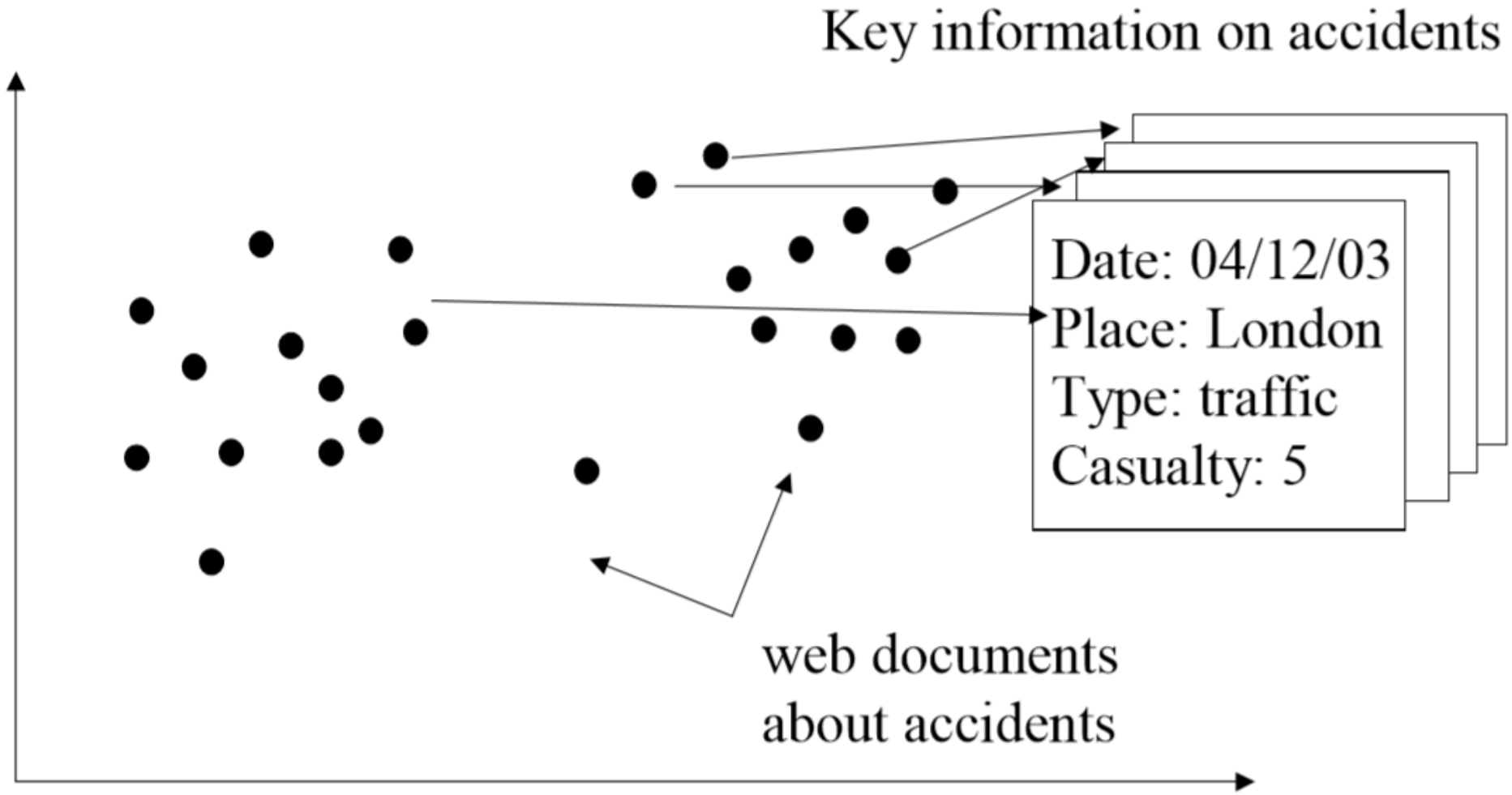
Information retrieval



Information filtering



Information extraction



Rule-based klasifikacija

- Napisati skup pravila po kojima se klasifikuju dokumenti
  - Visoka preciznost, lako održavanje sve dok je broj pravila mali
  - Problem, preklapanje pravila, konflikti, rekonstruisanje pravila kada se mijenja domen

$$\text{“ball”} \in d \rightarrow t(d) = \text{sports}$$

$$\text{“ball”} \in d \ \& \ \text{“dance”} \notin d \rightarrow t(d) = \text{sports}$$

$$\text{“ball”} \in d \ \& \ \text{“dance”} \notin d \ \& \ \text{“game”} \in d \ \&$$
$$\text{“play”} \in d \rightarrow t(d) = \text{sports}$$

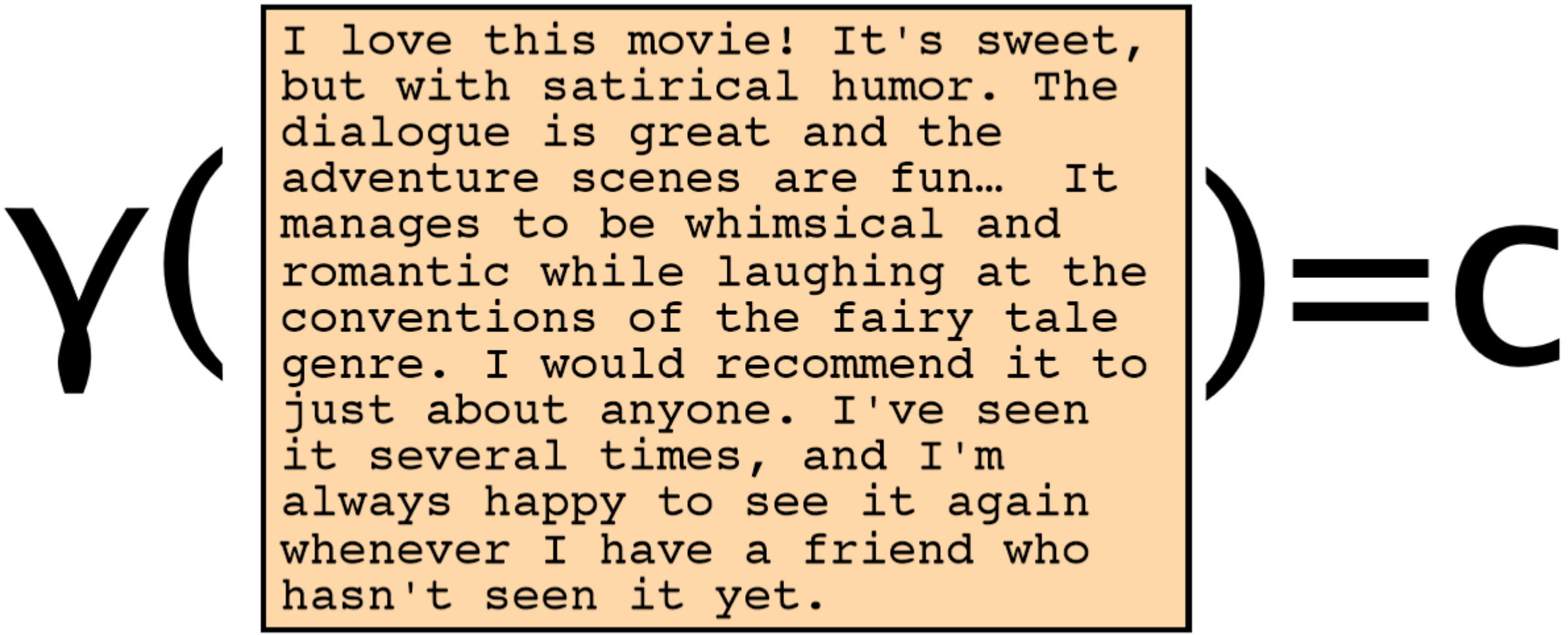
Machine learning based klasifikacija

- Nezavisna od domena
- Visoka tačnost
- Zahtijeva se postojanje skupa za obučavanje

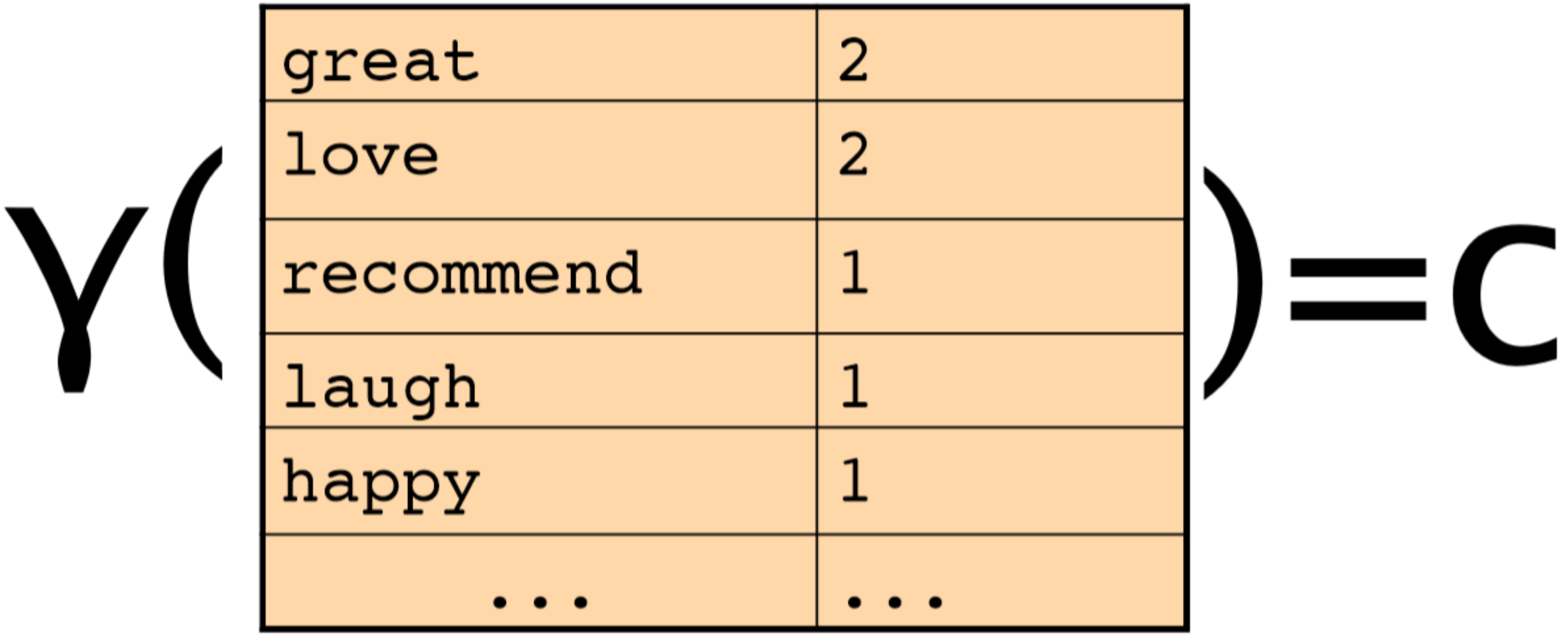
Formalna definicija problema

- Dat je skup dokumenata D={d1,d2,…,dn}
- Dat je skup klasa K={k1,k2,….,km}
- Potrebno je odrediti funkciju t:D->K, tako da je t(d) klasa dokumenta d

Klasifikacija dokumenata

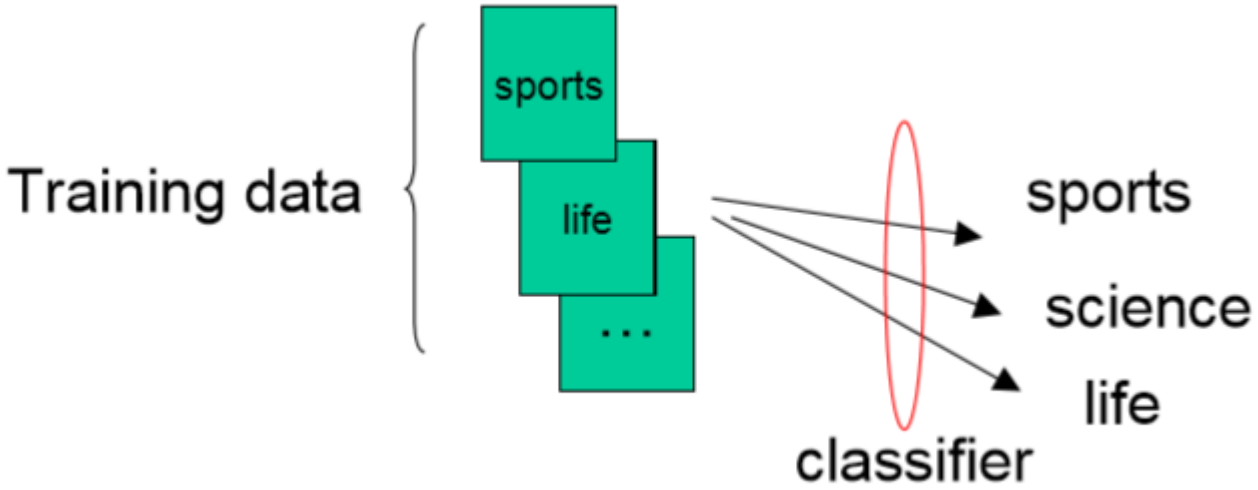


Bag-of-words



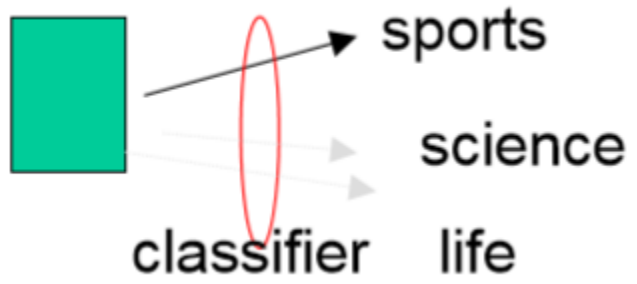
Machine learning approach

- Faza treniranja

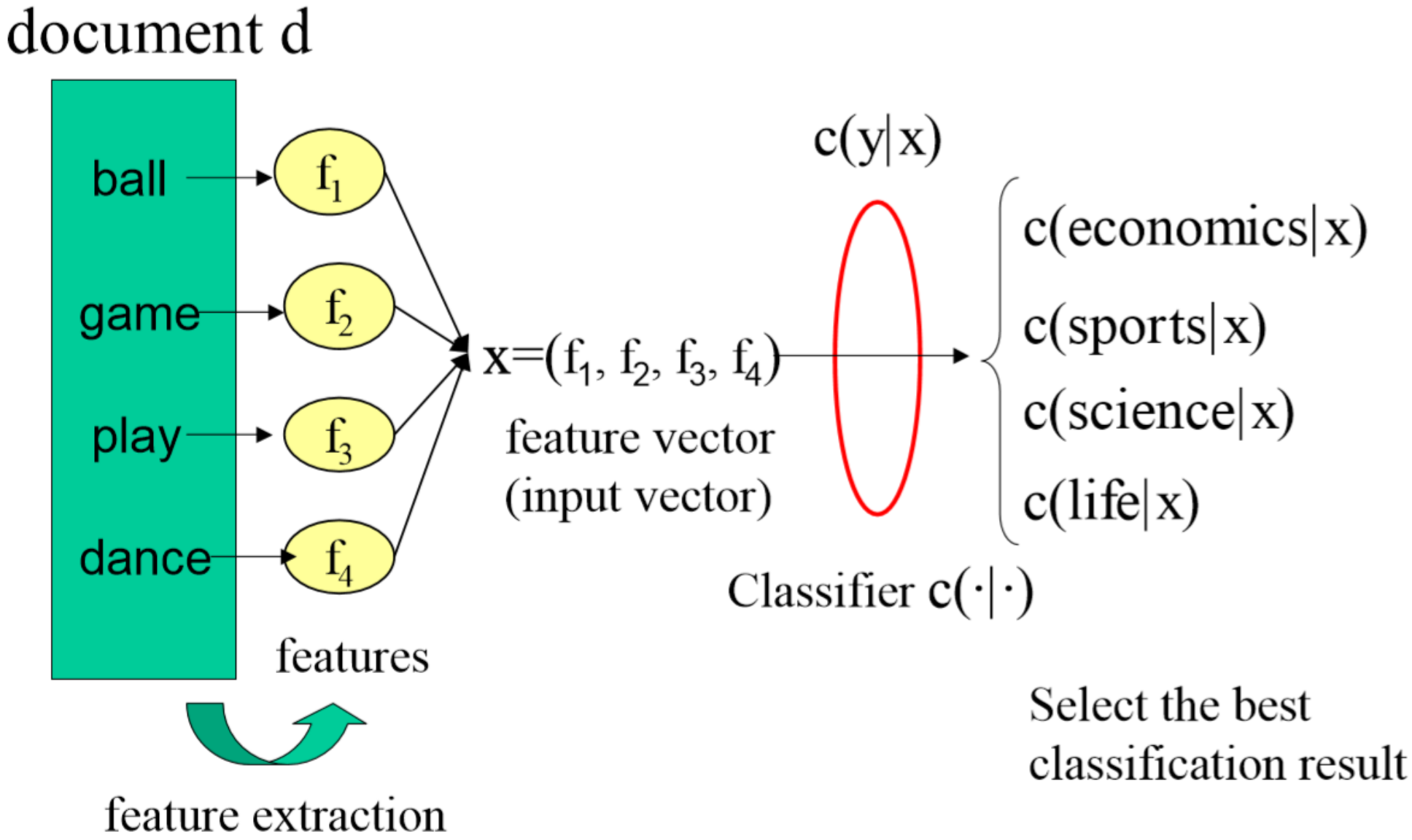


- Konstruisanje klasifikatora/modela primjenom algoritama mašinskog učenja
  - SVM, NB, DT, NN, LR itd.

- Klasifikacija novog dokumenta sa modelom



Machine learning approach (2)

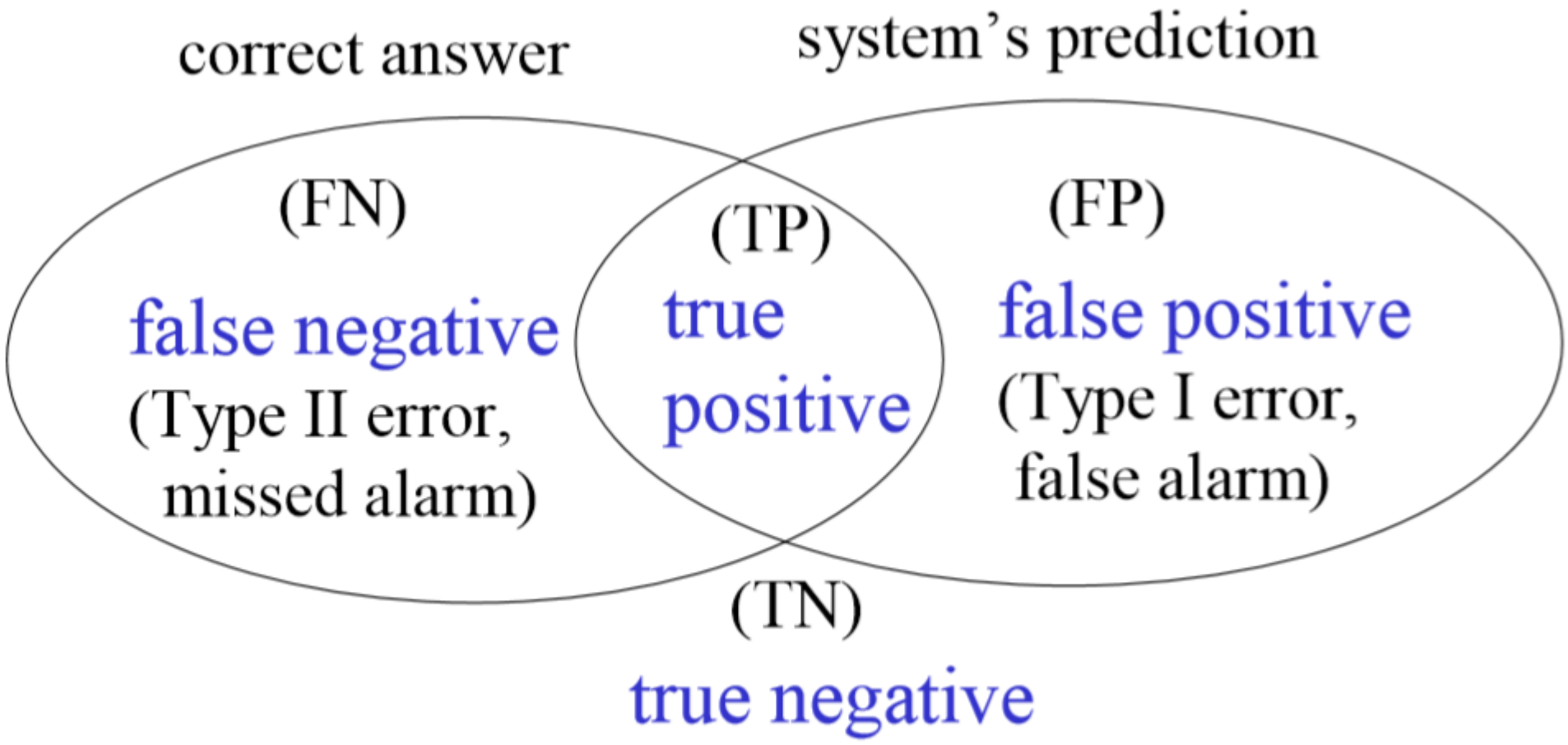


Procjena performansi

- Accuracy
- Precision
- Recall
- F-measure

	correct	not correct
selected	tp	fp
not selected	fn	tn

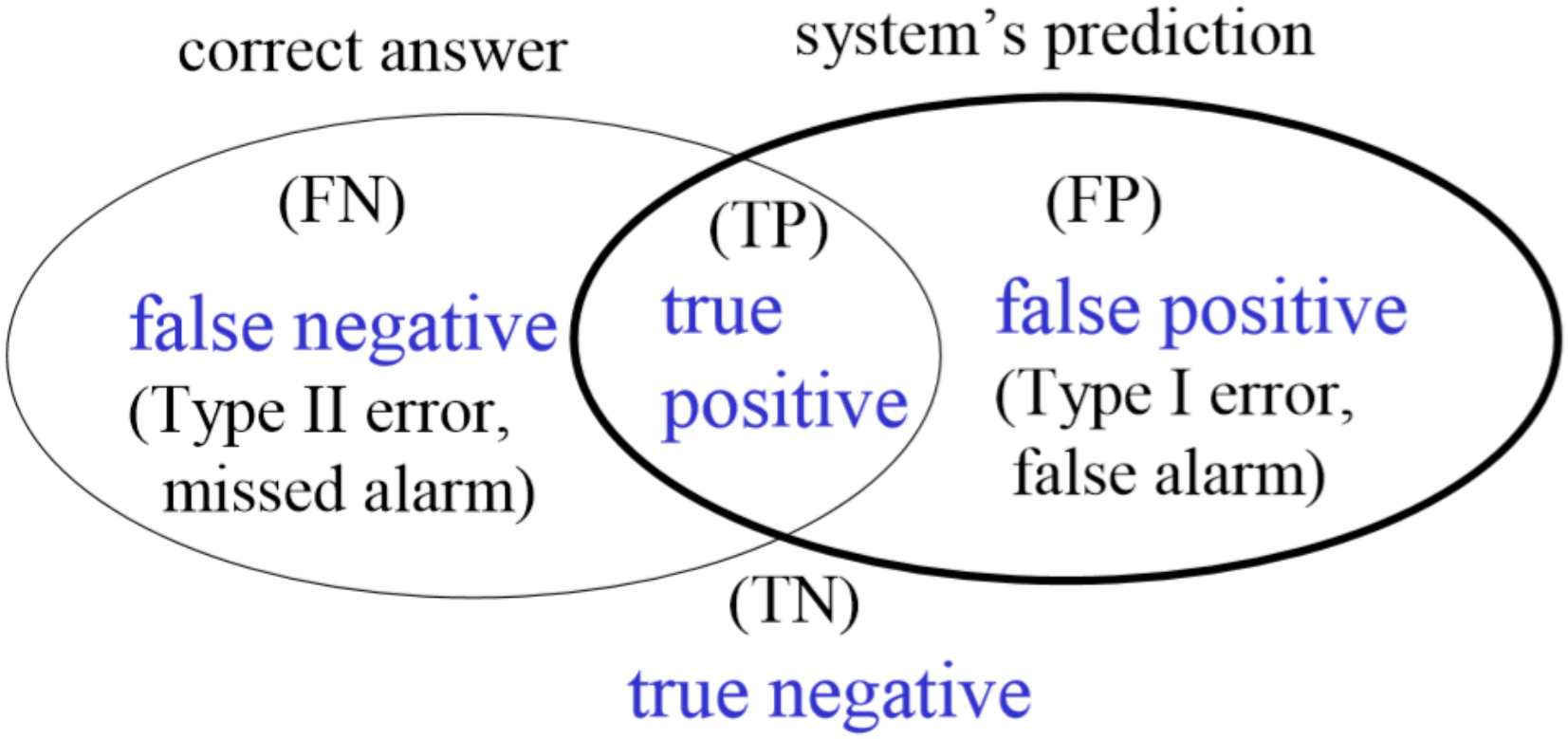
Accuracy



$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision

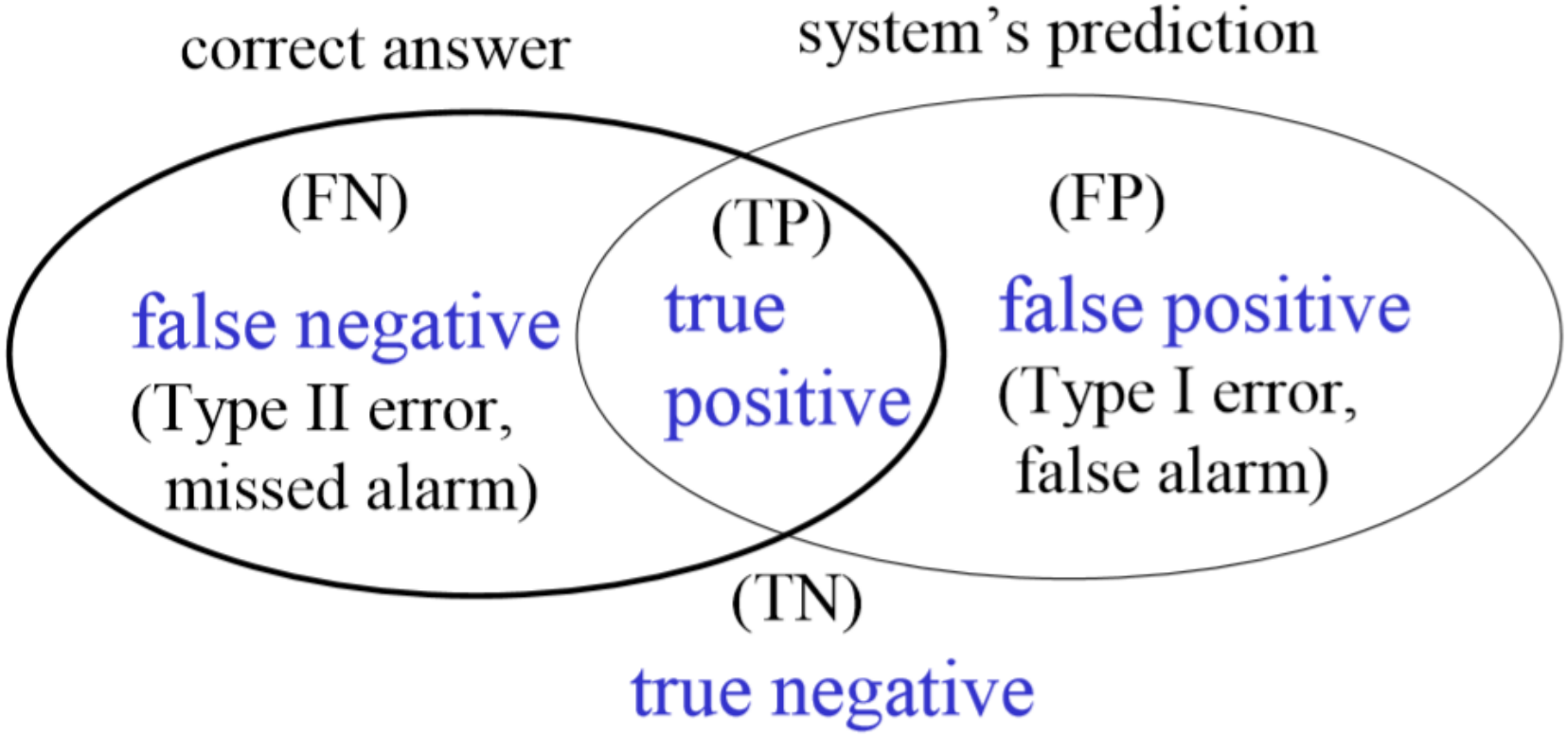
- The rate of correctly predicted topics



$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

- The rate of correctly predicted topics



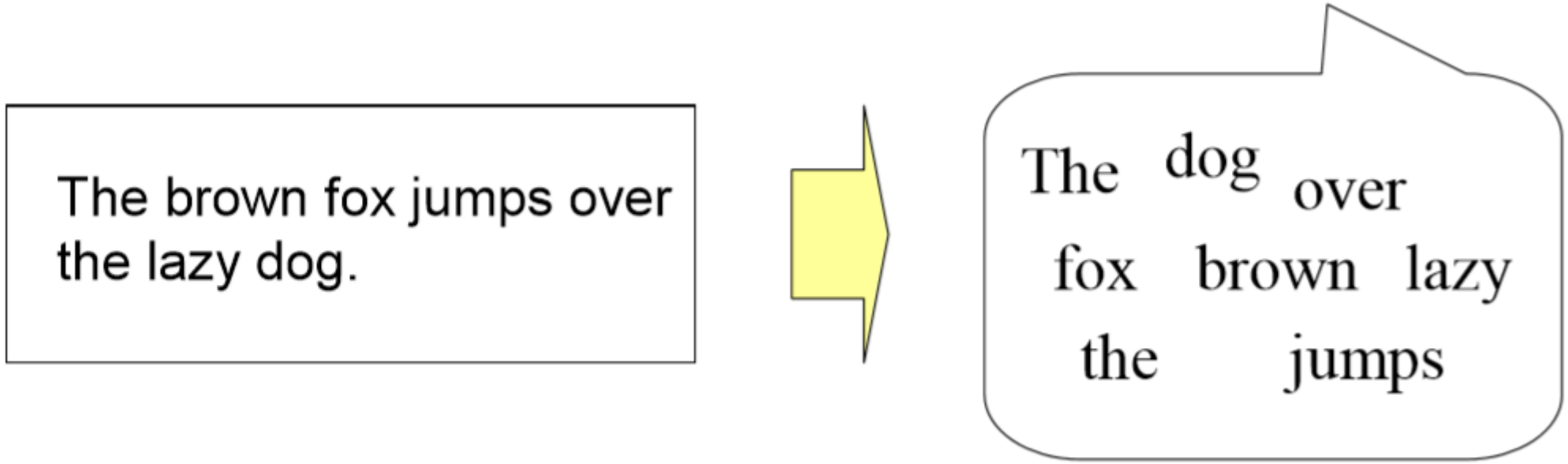
$$\text{Recall} = \frac{TP}{TP + FN}$$

F-measure

$$\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Predstavljanje tekstualnih dokumenata

- Bag-of-Word pristup
  - Dokument se posmatra kao skup riječi bez informacija o poretku i gramatici



Bag-of-Words

- Bi-gram, Tri-gram, n-gram, shingling

The brown fox jumps over the lazy dog.

Bi-grams:

the brown,
brown fox,
fox jumps,
jumps over,
the lazy,
lazy dog

Tri-grams:

the brown fox,
brown fox jumps,
fox jumps over,
jumps over the,
the lazy dog

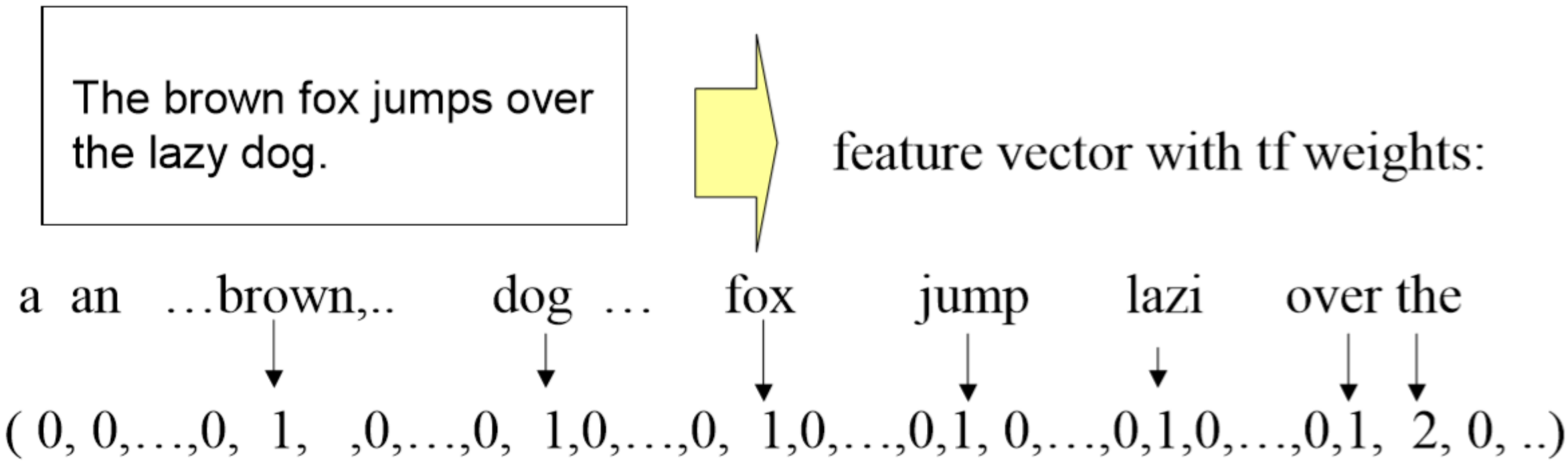
Bag-of-words (2)

- Normalizacija
  - Down-case
  - Lemmatizacija
    - Koristi se samo korijen riječi
  - Stemming
    - if the word ends in 'ed', remove the 'ed'
    - if the word ends in 'ing', remove the 'ing'
    - if the word ends in 'ly', remove the 'ly’

Bag-of-Words (3)

- Binarna reprezentacija, 1/0
- Term frequency
  - tf_i – broj pojavljivanja riječi/n-grama w_i u dokumentu
- Inverse document frequency
  - $idf_i = |D|/|\{d|d \text{ sadrži } w_i\}|$
- tf-idf
  - $tf\text{-}idf_i = tf_i * idf_i$
  - Frekventne riječi koje se pojavljuju u malom broju dokumenata dobiju visoki tf-idf

Vektor svojstava



Implementacija

```
def build_classifier(dir_name, test_dir_name):

    classes = ['k1', 'k2']
    train_data = load_files(dir_name, classes)
    count_vect   = CountVectorizer(decode_error = 'ignore')
    train_counts = count_vect.fit_transform(train_data.data)
    tfidf_transformer = TfidfTransformer()
    train_tfidf = tfidf_transformer.fit_transform(train_counts)
    classifier = SGDClassifier(max_iter=1000, loss='log').fit(train_tfidf, train_data.target)

    test_data       = load_files(test_dir_name, classes)
    test_counts     = count_vect.transform(test_data.data)
    test_tfidf      = tfidf_transformer.transform(test_counts)
    predicted = classifier.predict(test_tfidf)

    acc = round(np.mean(predicted == test_data.target) * 100, 2)
    print("Accuracy = " + str(acc))
```

Sistem za analizu dokumenata

1. Klasifikacija, information extraction
2. Rad sa kolekcijama dokumenata
    a. Podržani tipovi
        i. txt
        ii. searchable pdf (pomoću biblioteke PdfMiner)
        iii. images (pomoću Tesseract)
3. Pre-procesiranje
4. Reprezentacija dokumenata
5. Treniranje i testiranje modela
6. Upravljanje napravljenim modelom
7. Analiza novog skupa dokumenata
    a. Batch
    b. Stream
8. Eksperiment