Αναπαράσταση κειμένων και προεπεξεργασία

Ηρακλής Βαρλάμης

Περιεχόμενα

- Μοντέλα αναπαράστασης κειμένων
- Προεπεξεργασία κειμένων και ιστοσελίδων
- Αναζήτηση στο Web

Υπόθεση εργασίας

Υποθέτουμε ότι έχουμε μια συλλογή από κείμενα:

- με τα οποία θέλουμε να εκπαιδεύσουμε έναν ταξινομητή (classification)
- τα οποία θέλουμε να ομαδοποιήσουμε (clustering)
- από τα οποία θέλουμε να παράξουμε άλλα κείμενα (μεταφράσεις, συνόψεις, κλπ.)
- από τα οποία θέλουμε να εξάγουμε σημαντική πληροφορίας (named entity recognition, information extraction)

Μοντέλα αναπαράστασης

- Καθορίζουν την αναπαράσταση ενός κειμένου
- Επηρεάζουν στη συνέχεια τον τρόπο ορισμού της ομοιότητας/σχετικότητας μεταξύ κειμένων
- Κυριότερα μοντέλα
  - **Boolean model**
  - **Vector space model**
  - Probabilistic model
  - κλπ

Γενικό πλαίσιο μοντέλων αναπαράστασης

- Κάθε κείμενο είναι ένα τσουβάλι λέξεις (“bag” of words)
  - Η διάταξη και η θέση των λέξεων στην πρόταση ή στο έγγραφο αγνοείται
- Κάθε λέξη (όρος-term) σε κάθε έγγραφο συνδέεται με ένα βάρος (weight)
  - Θεωρητικά μπορούμε να χρησιμοποιήσουμε και αρνητικά βάρη
  - Συνήθως τα βάρη είναι θετικά. Αν ο όρος t_i δεν υπάρχει σε ένα κείμενο τότε το βάρος του είναι $w_i = 0$

Boolean model

- Αναπαράσταση εγγράφου:
  - Για μια συλλογή εγγράφων *D*, και το σύνολο

    $V = \{t_1, t_2, ..., t_{|V|}\}$ των |V| διακριτών όρων που εμφανίζονται στη συλλογή. Το *V* είναι το λεξικό (vocabulary).

  - $w_{ij} = 1$, αν το t_i εμφανίζεται στο $\mathbf{d}_j$
  - $w_{ij} = 0$, διαφορετικά
- Κάθε συλλογή εγγράφων αναπαρίσταται ως πίνακας

π.χ.

d1: **η** επεξεργασία φυσικής γλώσσας **ασχολείται με κείμενα**

d2: **η** λογοτεχνία και **η** ποίηση **ασχολούνται με κείμενα**

d3: τα "**κείμενα**" **με** δυσκολεύουν σαν μάθημα

d4: σήμερα θα **ασχοληθούμε** με την **ανάγνωση κειμένων**

Ορίζουμε ένα χώρο με τόσες διαστάσεις όσες οι διακριτές λέξεις των κειμένων μας

	d1	d2	d3	d4
η	1	1	0	0
επεξεργασία	1	0	0	0
φυσικής	1	0	0	0
γλώσσας	1	0	0	0
ασχολείται	1	1	0	0
με	1	1	1	1
κείμενα	1	1	1	0
λογοτεχνία	0	1	0	0
ποίηση	0	1	0	0
ασχολούνται	0	1	0	0
τα	0	0	1	0
δυσκολεύουν	0	0	1	0
σαν	0	0	1	0
μάθημα	0	0	1	0
σήμερα	0	0	0	1
θα	0	0	0	1
ασχοληθούμε	0	0	0	1
την	0	0	0	1
ανάγνωση	0	0	0	1
κειμένων	0	0	0	1

Διανυσματική αναπαράσταση

- Vector representation
- Ένα **βάρος** $w_{ij} > 0$ μπαίνει για τον όρο t_i που εμφανίζεται στο έγγραφο $\mathbf{d}_j \in D$.
- Αν ο ίδιος όρος δεν εμφανίζεται στο έγγραφο $\mathbf{d}_j$ *τότε* $w_{ij} = 0$.

$$\mathbf{d}_j = (w_{1j}, w_{2j}, \ldots, w_{|V|j})$$

- Για τον υπολογισμό του βάρους χρησιμοποιούνται διάφορες παραλλαγές της συχνότητας εμφάνισης ενός όρου στο κείμενο (TF) και της συχνότητας εμφάνισης ενός όρους στα έγγραφα της συλλογής (IDF)

Term Frequency (TF)

- Το βάρος ενός όρου t_i στο έγγραφο $\mathbf{d}_j$ είναι ο αριθμός εμφανίσεων του όρου t_i στο $\mathbf{d}_j$, και συμβολίζεται f_{ij}
- Μπορούμε επίσης να κανονικοποιήσουμε τα βάρη
  - π.χ. διαιρώντας με το πλήθος των όρων του $\mathbf{d}_j$
  - διαιρώντας με το μέγιστο f_{ij} για το εκάστοτε $\mathbf{d}_j$
- Προβλήματα
  - Ορισμένες λέξεις είναι πολύ κοινές και δεν έχουν καμία πληροφοριακή αξία (π.χ. άρθρα, σύνδεσμοι κλπ – stopwords)
  - Ορισμένες λέξεις είναι πολύ κοινές για μια συλλογή (π.χ. η λέξη gene σε μια συλλογή βιολογικών κειμένων).

TF-IDF weighting scheme

- Το πιο δημοφιλές μοντέλο υπολογισμού βαρών
  - TF: καθαρή συχνότητα εμφάνισης όρου στο κείμενο (**term frequency**)
  - IDF: αντίστροφη συχνότητα εμφάνισης στα έγγραφα της συλλογής (**inverse document frequency**)

  N: πλήθος εγγράφων στη συλλογή

  df_i: πλήθος εγγράφων που περιέχουν το t_i
- Τελικά το TF-IDF βάρος είναι:

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

ή

$$tf_{ij} = \frac{f_{ij}}{len(d_j)}$$

$$idf_i = \log\frac{N}{df_i}$$

$$w_{ij} = tf_{ij} \times idf_i$$

Παράδειγμα

d1: **η** επεξεργασία φυσικής γλώσσας **ασχολείται με κείμενα**

d2: **η** λογοτεχνία και **η** ποίηση **ασχολούνται με κείμενα**

d3: τα “**κείμενα**” **με** δυσκολεύουν σαν μάθημα

d4: σήμερα θα **ασχοληθούμε** με την **ανάγνωση κειμένων**

TF(η, d1)= 1/7, TF(η, d2)=2/8, TF(η, d3)=0, TF(η, d4)=0

IDF(η) =log 4/2

TF(κείμενα, d1)= 1/7, TF(κείμενα, d2)=1/8, TF(κείμενα, d3)=⅙, TF(κείμενα, d4)=0/7

IDF(κείμενα) =log 4/3

	d1	**d2**	**d3**	**d4**
η	1/7*log4/2	2/8*log4/2	0	0
επεξεργασία				
φυσικής				
γλώσσας				
ασχολείται				
με				
κείμενα	1/7*log4/3	2/8*log4/3	1/6*log4/3	0
λογοτεχνία				
ποίηση				
ασχολούνται				
τα				
δυσκολεύουν				
σαν				
μάθημα				
σήμερα				
θα				
ασχοληθούμε				
την				
ανάγνωση				
κειμένων				

Ανάκτηση στο vector space model

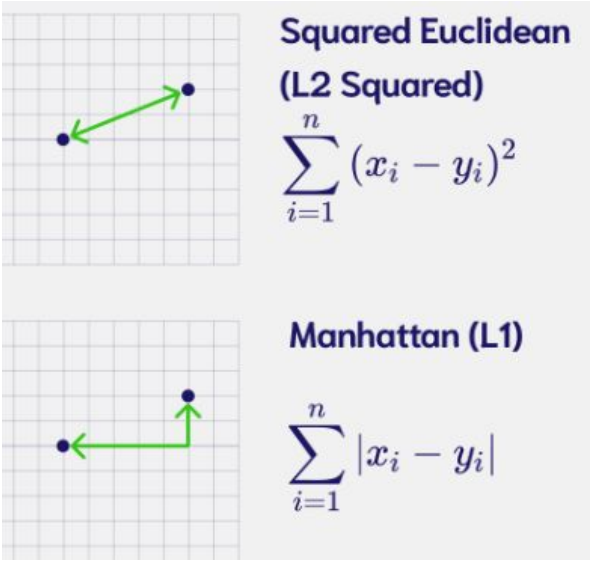
- Το ερώτημα **q** αναπαρίσταται με παρόμοιο τρόπο
- Η σχετικότητα (ομοιότητα) του d_j ως προς το q:
- Ομοιότητα συνημιτόνου - Cosine similarity
  - το συνημίτονο της γωνίας μεταξύ των δύο διανυσμάτων (d_j και q)

$$cosine(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\| \mathbf{d}_j \| \times \| \mathbf{q} \|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}$$

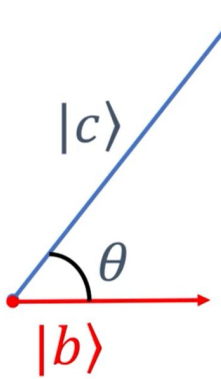
- Το Cosine χρησιμοποιείται και στο text clustering
- Βαθμονόμηση – Ranking: Με βάση τα σκορ ομοιότητας

Εναλλακτικά μέτρα ομοιότητας

- Γινόμενο (dot product): αντίστοιχο του cosine χωρίς κανονικοποίηση
- Ευκλείδεια απόσταση
- Manhattan απόσταση



$$\vec{v}.\vec{w} = \sum_{i=1}^{n} v_i.w_i$$



$$\vec{c} \cdot \vec{b} = |\vec{c}||\vec{b}|\cos\theta$$

Παράδειγμα

- Ένα λεξικό με τρεις όρους:
  - V = {hardware, software, users}
- Μια συλλογή 9 κειμένων:
  - A1=(1, 0, 0), A2=(0, 1, 0), A3=(0, 0, 1)
  - A4=(1, 1, 0), A5=(1, 0, 1), A6=(0, 1, 1)
  - A7=(1, 1, 1), A8=(1, 0, 1). A9=(0, 1, 1)
- Αν το ερώτημα είναι “hardware και software” ποια κείμενα θα ανακτηθούν;
  - Boolean query matching
  - Vector space matching

Παράδειγμα

- Boolean query matching:
  - Τα έγγραφα A4, A7 θα ανακτηθούν ως τα μόνα που έχουν ΚΑΙ τους δύο όρους
- Similarity matching (cosine):
  - q=(1, 1, 0)
  - S(q, A1)=0.71
  - S(q, A2)=0.71
  - S(q, A3)=0
  - S(q, A4)=1
  - S(q, A5)=0.5
  - S(q, A6)=0.5
  - S(q, A7)=0.82
  - S(q, A8)=0.5
  - S(q, A9)=0.5
  - Το βαθμονομημένο σύνολο αποτελεσμάτων θα περιλαμβάνει=
    - {A4, A7, A1, A2, A5, A6, A8, A9}

Μέτρα επικάλυψης για το Bag of words

$\lvert Q \cap D \rvert$	Simple matching (coordination level match)
$2\frac{\lvert Q \cap D \rvert}{\lvert Q \rvert + \lvert D \rvert}$	Dice's Coefficient
$\frac{\lvert Q \cap D \rvert}{\lvert Q \cup D \rvert}$	Jaccard's Coefficient
$\frac{\lvert Q \cap D \rvert}{\lvert Q \rvert^{1/2} \times \lvert D \rvert^{1/2}}$	Cosine Coefficient
$\frac{\lvert Q \cap D \rvert}{\min(\lvert Q \rvert, \lvert D \rvert)}$	Overlap Coefficient

Προεπεξεργασία

- Πολύ απλή: απλά πέρασε το ερώτημα στο μηχανισμό ανάκτησης
  - Προεπεξεργασία (π.χ. αφαίρεσε τα stop-words)
- Πιο σύνθετη
  - Μετέτρεψε ένα ερώτημα φυσικής γλώσσας σε ερώτημα προς το μηχανισμό ανάκτησης
  - Χρησιμοποίησε user feedback για να βελτιώσεις (επεκτείνεις/εκλεπτύνεις) το ερώτημα (relevance feedback)
  - κλπ..

Μηχανισμός ανάκτησης

- Υπολογίζει ένα βαθμό σχετικότητας (relevance score)
  - Για κάθε έγγραφο
  - Ως προς το ερώτημα
- Οι βαθμοί αυτοί χρησιμοποιούνται για να βαθμονομήσουμε (rank) τα έγγραφα πριν τα δείξουμε στο χρήστη
- Συνήθως μόνο ένα μικρό υποσύνολο εγγράφων συγκρίνεται (και βαθμολογείται) ως προς το ερώτημα

Περιεχόμενα

- Μοντέλα αναπαράστασης κειμένων
- Προεπεξεργασία κειμένων και ιστοσελίδων
- Αναζήτηση στο Web

Προεπεξεργασία κειμένου

- Εντοπισμός λέξεων: tokenization
- Εντοπισμός όρων
- Αφαίρεση stopwords
- Αποκοπή ρίζας (stemming)
- Υπολογισμός συχνοτήτων εμφάνισης και TF-IDF βαρών.

Stopwords

- Λέξεις που δεν μεταφέρουν κάποια έννοια, αποτελούν θόρυβο και πρέπει να αφαιρεθούν (“stopped”) πριν ευρετηριαστούν
  - π.χ. : και, το, ο, η, στο, ή, σε, για, κλπ…
- Ο ρόλος τους στο κείμενο είναι καθαρά συντακτικός
- Κρατούνται σε ένα αρνητικό λεξικό
  - 10–1000 λέξεις
  - Π.χ. http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words
- Για ένα συγκεκριμένο πεδίο εφαρμογής μπορεί να χρειαστούμε και μια επιπλέον εξειδικευμένη λίστα stopwords

Αφαίρεση stopwords

- Κατά την ανάγνωση και διάσπαση του κειμένου σε τμήματα (tokenization) τα ανιχνεύουμε και τα αφαιρούμε
- Γιατί τα αφαιρούμε;
  - Μειώνουμε το μέγεθος του ευρετηρίου
    - Τα stopwords αντιστοιχούν στο 20-30% του συνολικού αριθμού εμφανίσεων λέξεων
  - Βελτιώνουμε την απόδοση του συστήματος ανάκτησης
    - Τα stopwords δεν είναι χρήσιμα και μπορεί να μπερδέψουν το σύστημα ανάκτησης

Εύρεση ρίζας - Stemming

- Η βασική δομή μιας λέξης στη Νέα Ελληνική Γλώσσα αποτελείται από τα ακόλουθα τμήματα,

  Πρόθεμα + Ρίζα(Stem) + Κατάληξη + Τονισμό

  Σκοπός μας είναι να βρούμε την ρίζα κάθε λέξης

  ✔ Δηλαδή η ορθή αφαίρεση της κατάληξης κάθε λέξης (*suffix stripping*)

  ✔ Με τη Χρήση γλωσσικών μορφολογικών κανόνων

- Στόχος είναι να εντοπίσουμε τη ρίζα (root/stem) κάθε λέξης. π.χ.,

• χρήστες	απέχω
• χρήστης	απέχουμε
• χρηστικός	αποχή
stem: χρηστ	απεχ

Βασικές μέθοδοι stemming

Μια σειρά από κανόνες: π.χ. για αγγλικά

- Αφαίρεση κατάληξης
  - if a word ends with a consonant other than s,

    followed by an s, then delete s.
  - if a word ends in es, drop the s.
  - if a word ends in ing, delete the ing unless the remaining word consists only of one letter or of th.
  - If a word ends with ed, preceded by a consonant, delete the ed unless this leaves only a single letter.
- Μετασχηματισμός λέξης
  - if a word ends with “ies” but not “eies” or “aies” then “ies --> y.”

Μορφολογικοί Κανόνες

- Στη Νέα Ελληνική Γλώσσα συναντάμε: Ρήματα, Επίθετα, Ουσιαστικά, Επιρρήματα, Πρόθεσεις, Σύνδεσμοι, Επιφωνήματα κλπ.
- Το μεγαλύτερο πλήθος των λέξεων υπάγεται στα ρήματα – επίθετα – ουσιαστικά

Ρήματα	Επίπεδο 1	ώ/ω, ά, έ, ί, ύ, ή, εί ,αί, άω
	Επίπεδο 2	β-,γ-,ζ-,φ-,χ-,θ-,κ-,λ-αίν-, (ακολουθεί μια κατάληξη από το επίπεδο 1) -νω, -ήθηκα, -ύθηκα, -αίνομαι, -ημένος, -άνθηκα, -άχτηκα, -έρνω, -άνων, -αίω, -αίξα -αίχτηκα, -μένος, -όμαι, -άμαι, -ούμαι, -ιέμαι
Ουσιαστικά	Επίπεδο 1	ός/ος, άς/ας, ές/ες, ής/ης, ούς/ους, ως, οί/οι, ά/α ,έ/ε, ή/η, ό/ο, ί/ι
	Επίπεδο 2	ιμ, ουδ, ατ, ηδ, εδ, αδ
Επίθετα	Επίπεδο 1	ος, ης, ούι άς, ων, ά/α, έ/ε, ή/η, ό/ο, ί/ι, υ/ύ
	Επίπεδο 2	ικ, άδικ, ούδικ, ότερ, ότατ, ώτερ, ώτατ, ύτερ, ύτατ, έστερ, έστατ, ούσ/ουσ, ονας, ούλ, άσι, ιάρ, άρ, υμέν, υσμέν, αγμέν

Εργαλεία Stemming

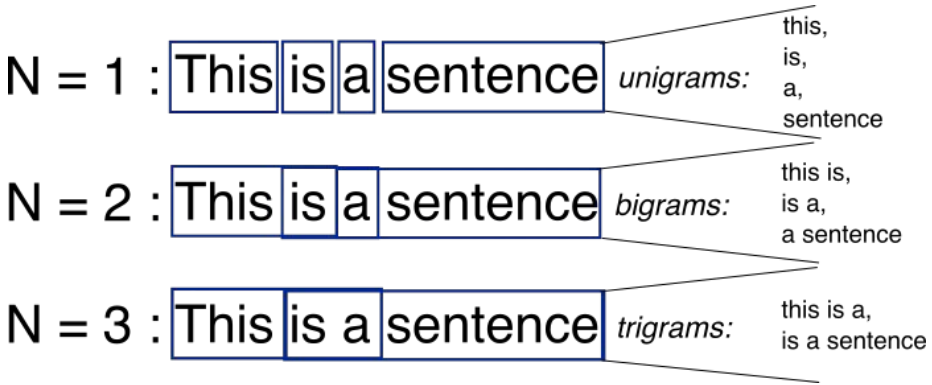
- Εξαρτώνται από τη γλώσσα
  - Porter stemmer για αγγλικά: http://www.tartarus.org/~martin/PorterStemmer/
  - Ο Porter έχει επίσης αναπτύξει το Snowball, μια γλώσσα για τη δημιουργία αλγορίθμων stemming σε οποιαδήποτε γλώσσα http://snowball.tartarus.org/
- https://pypi.org/project/greek-stemmer/

Οφέλη από το Stemming

- Βελτίωση της αποτελεσματικότητας του IR
  - Μειώνεται ο χώρος λέξεων (μέγεθος λεξικού) και αυξάνεται η πιθανότητα ταιριάσματος λέξεων
  - Αυξάνει την ανάκληση (recall)
- Μείωση του μεγέθους του ευρετηρίου
  - Μπορεί να οδηγήσει σε μείωση 40-50%.
- Μείωση την ακρίβειας: πολλά κείμενα που δεν περιέχουν τη λέξη που θέλουμε, αλλά μια διαφορετική λέξη που όμως δίνει το ίδιο stem επιστρέφονται.

Άλλες τεχνικές

- Λημματοποίηση - Lemmatization: Αντικατάσταση των λέξεων από το βασικό τους τύπο
  - am, are, is → be
  - car, cars, car's, cars' → car
  - Συνήθως χρειαζόμαστε κάποιο λεξικό ή θησαυρό ή κάποιο λεματοποιητή (αντίστοιχο του stemmer)
- Παραγωγή ν-γραμμάτων: Προσθήκη επιπλέον γνωρισμάτων που παράγονται από συνδυασμούς λέξεων (σε ένα κινούμενο παράθυρο)
- Sub-words and byte-pairs



Άλλες ενέργειες προεπεξεργασίας

- Αριθμοί
  - Στο IR: τους αφαιρούμε εκτός από πολύ συγκεκριμένους τύπους (χρηματικά ποσά, ημερομηνίες)
  - Στο Web: τους ευρετηριάζουμε
- Hyphens
  - Κάνουμε διάσπαση των λέξεων;
    - π.χ. «προ-επεξεργασία»
  - Συνήθως έχουμε ένα γενικό κανόνα και συγκεκριμένες εξαιρέσεις
- Punctuation Marks
  - Το ίδιο με τα hyphens
- Κεφαλαία/Μικρά
  - Συνήθως τα μετατρέπουμε όλα σε μια από τις δύο μορφές

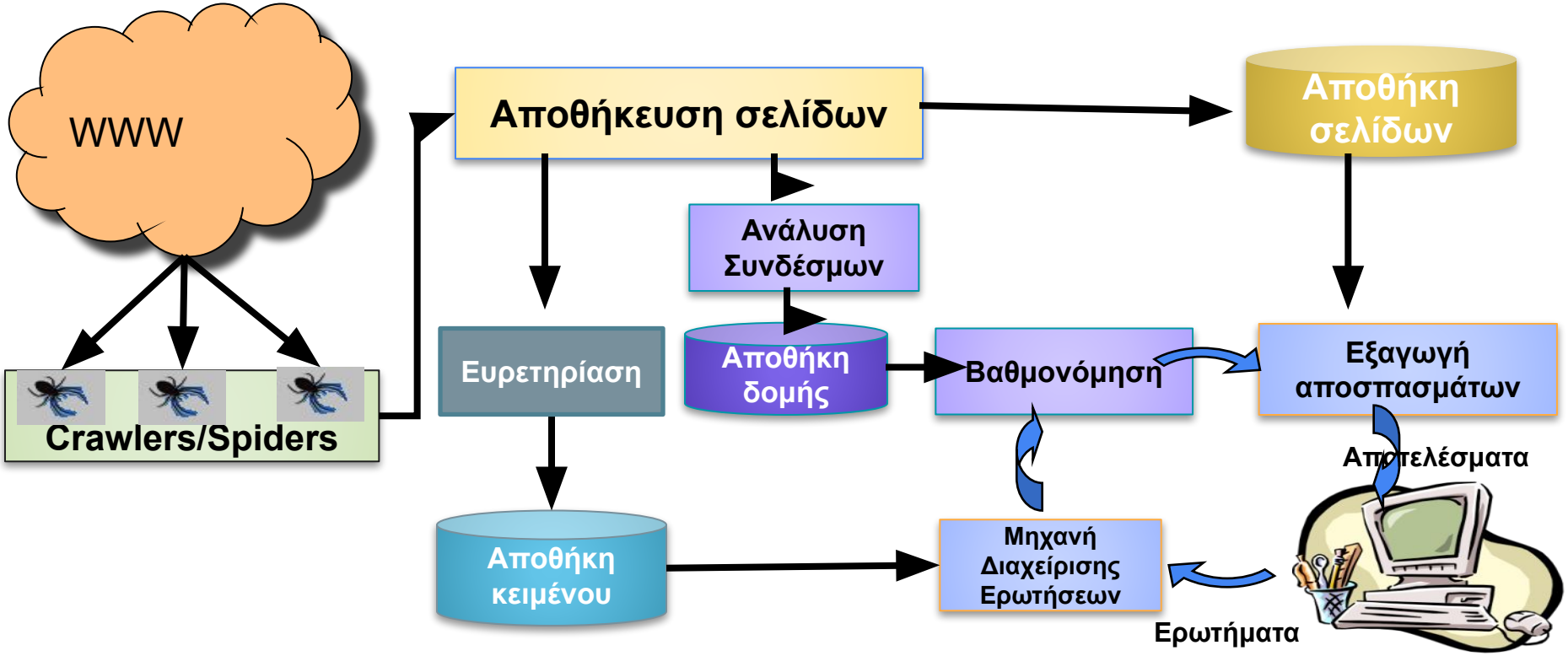
Περιεχόμενα

- Μοντέλα αναπαράστασης κειμένων
- Προεπεξεργασία κειμένων και ιστοσελίδων
- Αναζήτηση στο Web

Το Web Search ως ένα σύστημα IR

- Ένας Web crawler περιηγείται το Web και συλλέγει τις σελίδες που βρίσκει
- Τα αποτελέσματα αποθηκεύονται σε δομές γρήγορης αναζήτησης που λέγονται ανεστραμμένα ευρετήρια
- Την ώρα που υποβάλλεται ένα ερώτημα η μηχανή αναζήτησης κάνει μια σειρά από συγκρίσεις διανυσμάτων
- Οι σελίδες βαθμονομούνται με βάση
  - Την ομοιότητα περιεχομένου προς το ερώτημα (Similarity to query)
  - Τη δημοτικότητά τους Popularity (“Authority”)

Αρχιτεκτονική των Search Engines



Προεπεξεργασία ιστοσελίδων

- Εντοπισμός διαφορετικών πεδίων
  - π.χ. HTML: title, metadata, body
- Εντοπισμός του anchor text ενός link
  - Συχνά αυτό που λένε οι άλλοι για μια σελίδα είναι πιο περιγραφικό από αυτό που λέει η ίδια η σελίδα
- Αφαίρεση των HTML tags
  - Όμοια με το punctuation
- Εντοπισμός των μπλοκ περιεχομένου
  - Τμηματοποίηση οπτική με βάση τη δομή
    - Θέλει εκπαίδευση και χειροκίνητη επισήμανση
  - Ταίριασμα υποδέντρων
    - Πολλές ιστοσελίδες βασίζονται σε templates
    - Βρίσκει τα κρυμμένα templates

Υπολογισμός του TF-IDF

- TF: Υπολογίζουμε τον αριθμό εμφανίσεων ενός όρου στο έγγραφο
  - Η συχνότητα εμφάνισης είναι ανάλογη της σημαντικότητας του όρου για το έγγραφο
  - Αν ο όρος εμφανίζεται συχνά στο έγγραφο, τότε το έγγραφο μάλλον ασχολείται με θέματα σχετικά με τον όρο
- IDF: Μετρά τον αριθμό των εγγράφων της συλλογής που περιέχουν τον κάθε όρο
  - Προϋποθέτει ότι έχουμε στη διάθεσή μας όλη τη συλλογή από την αρχή
  - Αυτό δεν ισχύει στο web