# Filtrado Basado en Contenido

## IIC 3633 - Sistemas Recomendadores

Denis Parra
Profesor Asistente, DCC, PUC CHile

# TOC

## En esta clase

# Por Qué un Recomendador Basado en Contenido

- El filtrado colaborativo tiene algunas desventajas: cold-start, sparcity, transparency.

**PROS**

- A diferencia del Filtrado Colaborativo, si los items tienes descripciones suficientes, nos evitamos el "new-item problem""
- Las representaciones del contenido son variadas y permiten utilizar diversas técnicas de procesamiento del texto, uso de información semántica, inferencias, etc.
- Es sencillo hacer un sistema más transparente: usamos el mismo contenido para explicar las recomendaciones.

**CONS**

- Tienden a la sobre-especialización: va a recomendar items similares a los ya consumidos, creando una tendencia al "filter bubble".
- Los métodos basados en filtrado colaborativo han mostrado ser, empíricamente, más precisos al momento de generar recomendaciones.

# Arquitectura de un Sistema de Recomendación CB

- Los componentes principales son: (1) Analizador del Contenido, (2) Aprendizaje del Perfil de Usuario, (3) Filtrado de Contenido

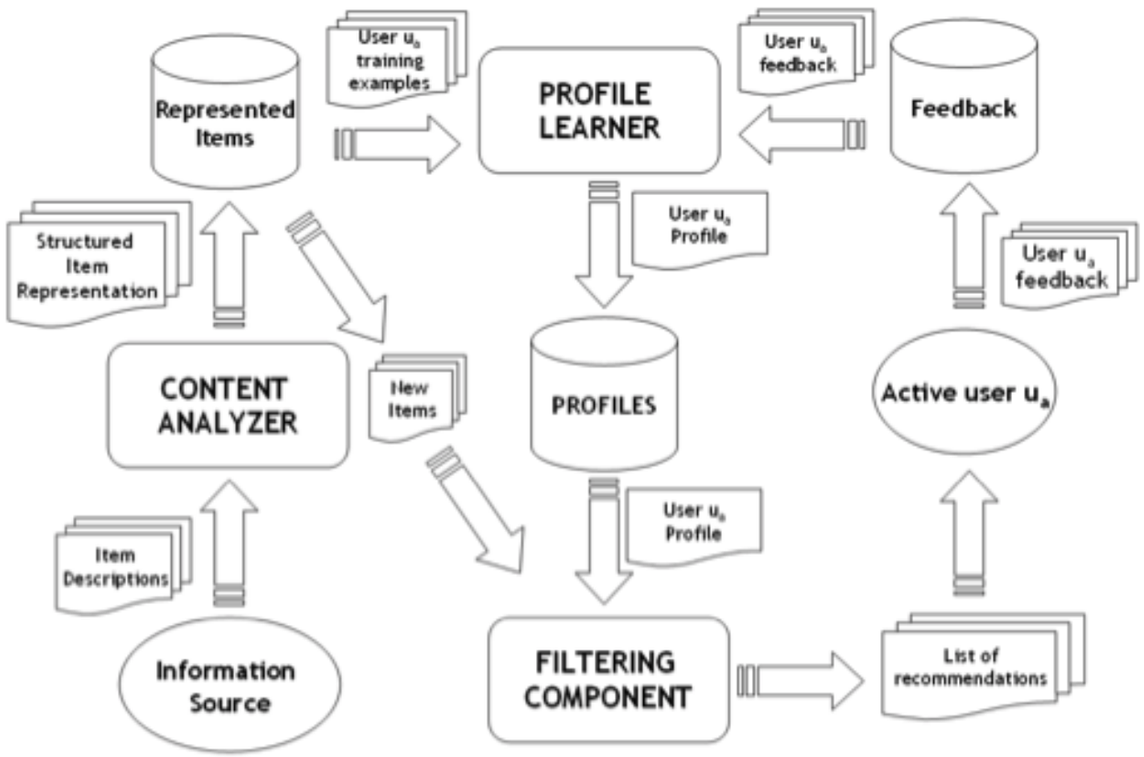
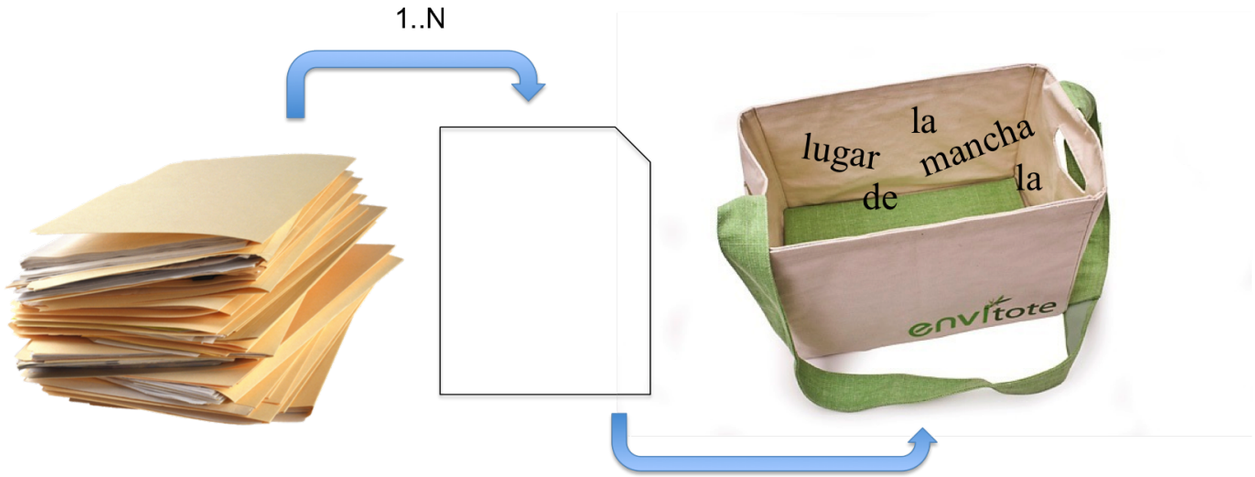



**Fig. 3.1:** High level architecture of a Content-based Recommender

# Representación del Contenido: Bolsa de Palabras

- Se suele representar a los documentos como "bolsas de palabras"; de esta forma es fácil pasar a representar cada documento como un vector (Vector Space Model)



# Representación del Contenido: VSM

- El corpus completo puede entonces representarse como una matriz donde las filas son términos y las columnas son documentos.

| | Anthony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth . . . |
|---|---|---|---|---|---|---|
| ANTHONY | 1 | 1 | 0 | 0 | 0 | 1 |
| BRUTUS | 1 | 1 | 0 | 1 | 0 | 0 |
| CAESAR | 1 | 1 | 0 | 1 | 1 | 1 |
| CALPURNIA | 0 | 1 | 0 | 0 | 0 | 0 |
| CLEOPATRA | 1 | 0 | 0 | 0 | 0 | 0 |
| MERCY | 1 | 0 | 1 | 1 | 1 | 1 |
| WORSER | 1 | 0 | 1 | 1 | 1 | 0 |
| . . . | | | | | | |

- Luego, ¿Cúal es la mejor forma de representar los pesos de los términos?

# Representación del Contenido: VSM II

## Frecuencia de los términos

Cada documento se representa como un vector, el "peso" de cada palabra para ese documento

$$\mathrm{TF}(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}}$$

puede darse en base a la frecuencia del término en el documento.

| | Anthony and Cleopatra | Julius Caesar | The Tempest | Hamlet | Othello | Macbeth . . . |
|---|---|---|---|---|---|---|
| ANTHONY | 157 | 73 | 0 | 0 | 0 | 1 |
| BRUTUS | 4 | 157 | 0 | 2 | 0 | 0 |
| CAESAR | 232 | 227 | 0 | 2 | 1 | 0 |
| CALPURNIA | 0 | 10 | 0 | 0 | 0 | 0 |
| CLEOPATRA | 57 | 0 | 0 | 0 | 0 | 0 |
| MERCY | 2 | 0 | 3 | 8 | 5 | 8 |
| WORSER | 2 | 0 | 1 | 1 | 1 | 5 |
| . . . | | | | | | |

Podemos normalizar el valor en función de la frecuencia máxima de cualquier término en el documento.

# Representación del Contenido: VSM III

## Log de Frecuencia de los términos

Pero el hecho que un término $x$ aparece 100 veces y otro término $y$ sólo 10 veces, no hace a x 10 veces más relevantes; por lo tanto podemos usar un logaritmo.

- La log-frecuencia del término t en d se define como

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d} & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

- $\mathrm{tf}_{t,d} \rightarrow w_{t,d}$ :

  $0 \rightarrow 0,\ 1 \rightarrow 1,\ 2 \rightarrow 1.3,\ 10 \rightarrow 2,\ 1000 \rightarrow 4,$ etc.

# Representación del Contenido: VSM IV

## TF-IDF

Bajo la intuición de que un término que aparece en sólo unos poco documentos podría ser descriptivo, podemos considerar la "Inverse Document Frequency" y combinarla con la "Term Frequency":

$$\text{TF-IDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{log \frac{N}{n_k}}_{\text{IDF}}$$

Done $t_k$ es el término $k$, $d_j$ es el documento $j$.

# Resumen de Componentes del TF-IDF

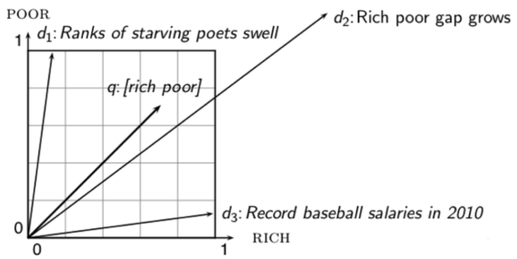
| Term frequency | | Document frequency | | Normalization | |
|---|---|---|---|---|---|
| n (natural) | $\mathrm{tf}_{t,d}$ | n (no) | 1 | n (none) | 1 |
| l (logarithm) | $1 + \log(\mathrm{tf}_{t,d})$ | t (idf) | $\log \frac{N}{\mathrm{df}_t}$ | c (cosine) | $\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$ |
| a (augmented) | $0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$ | p (prob idf) | $\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$ | u (pivoted unique) | $1/u$ |
| b (boolean) | $\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$ | | | b (byte size) | $1/CharLength^{\alpha}$, $\alpha < 1$ |
| L (log ave) | $\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$ | | | | |

# Representación Semántica del Contenido

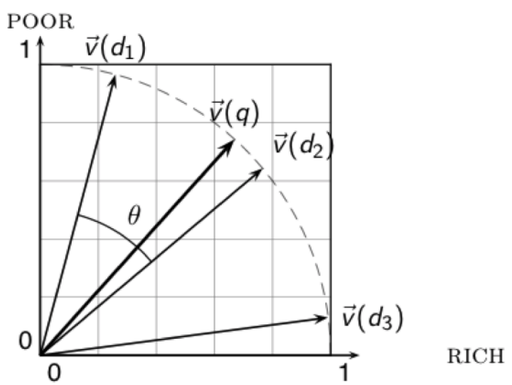
- No todo el contenido del documento corresponde a la misma categoría.
- Autor, palabras clave, fechas, tópicos pueden dar una noción adicional de filtrado.
- Opción 1: Representación semántica explícita (No lo veremos en detalle en esta clase)
  - Ontologías
  - WordNet
  - ConceptNet
- Opción 2: Inferir representación semántica (LSI, LDA)
- Opción 3: Word Vectors (Word2Vec, Glove)

# Buscando Items Similares

## Distancia Euclidiana

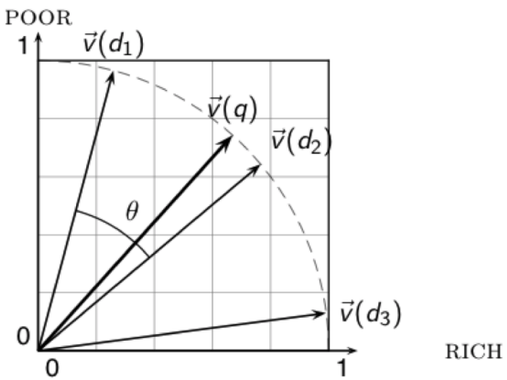



## Distancia Coseno



# Buscando Items Similares

## Distancia Coseno




## Fórmula

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k {w_{ki}}^2} \cdot \sqrt{\sum_k {w_{kj}}^2}}$$

# Buscando Items Similares II

## Okapi BM25

$$RSV_d = \sum_{t \in q} IDF \cdot \frac{(k_1 + 1) tf_{td}}{k_1((1-b) + b \times (L_d / L_{ave})) + tf_{td}} \cdot \frac{(k_3 + 1) tf_{tq}}{k_3 + tf_{tq}}$$

Ref: Denis Parra and Peter Brusilovsky. 2009. Collaborative filtering for social tagging systems: an experiment with CiteULike. In Proceedings of the third ACM conference on Recommender systems (RecSys '09) http://doi.acm.org/10.1145/1639714.1639757

# Buscando Items Similares III

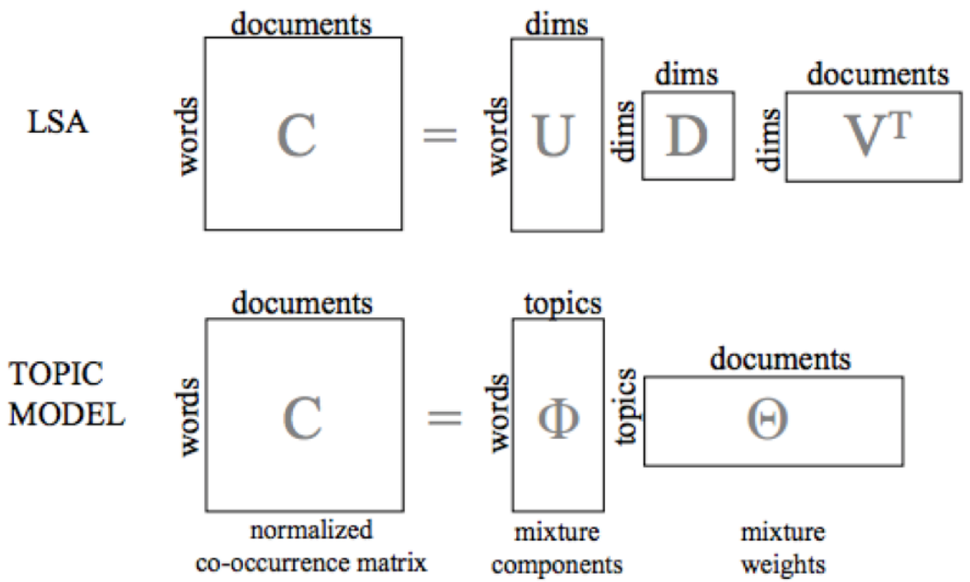
## Técnicas de Procesamiento adicionales

- Pasar a mayúsculas/minúsculas
- Tokenization
- Stemming (Porter, Krovetz)
- Lemmatization

# Buscando Items Similares

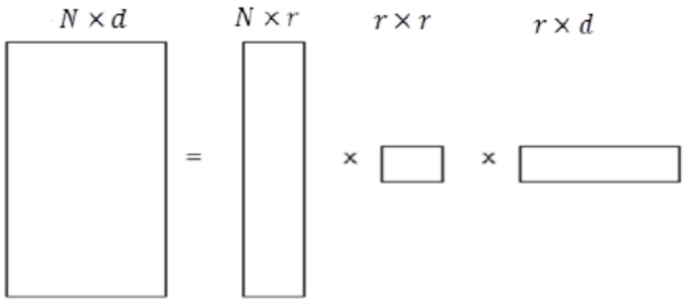
## Representación en espacio latente

- Latent Semantic Indexing
- Latent Dirichlet Allocation



# LSI I

$$\mathbf{X} = \mathbf{U} \times \mathbf{\Sigma} \times \mathbf{V^T}$$




$$(\mathbf{t}_i^T) \rightarrow \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} = (\hat{\mathbf{t}}_i^T) \rightarrow \begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} \cdot \begin{bmatrix} [ & \mathbf{v}_1 & ] \\ & \vdots & \\ [ & \mathbf{v}_l & ] \end{bmatrix}$$

$X$ $(\mathbf{d}_j)$ ↓ — $U$ — $\Sigma$ — $V^T$ $(\hat{\mathbf{d}}_j)$ ↓

# LSI II

| | | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ | |
|---|---|---|---|---|---|---|---|---|
| | ship | 1 | 0 | 1 | 0 | 0 | 0 | |
| | boat | 0 | 1 | 0 | 0 | 0 | 0 | |
| | ocean | 1 | 1 | 0 | 0 | 0 | 0 | |
| | voyage | 1 | 0 | 0 | 1 | 1 | 0 | |
| | trip | 0 | 0 | 0 | 1 | 0 | 1 | |

# LSI III

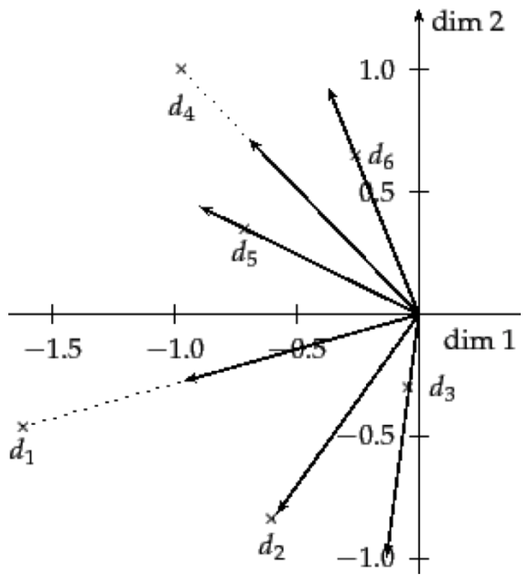
| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| ship | −0.44 | −0.30 | 0.57 | 0.58 | 0.25 |
| boat | −0.13 | −0.33 | −0.59 | 0.00 | 0.73 |
| ocean | −0.48 | −0.51 | −0.37 | 0.00 | −0.61 |
| voyage | −0.70 | 0.35 | 0.15 | −0.58 | 0.16 |
| trip | −0.26 | 0.65 | −0.41 | 0.58 | −0.09 |

$$\begin{bmatrix} 2.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.59 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.28 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.39 \end{bmatrix}$$

| | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ |
|---|---|---|---|---|---|---|
| 1 | −0.75 | −0.28 | −0.20 | −0.45 | −0.33 | −0.12 |
| 2 | −0.29 | −0.53 | −0.19 | 0.63 | 0.22 | 0.41 |
| 3 | 0.28 | −0.75 | 0.45 | −0.20 | 0.12 | −0.33 |
| 4 | 0.00 | 0.00 | 0.58 | 0.00 | −0.58 | 0.58 |
| 5 | −0.53 | 0.29 | 0.63 | 0.19 | 0.41 | −0.22 |

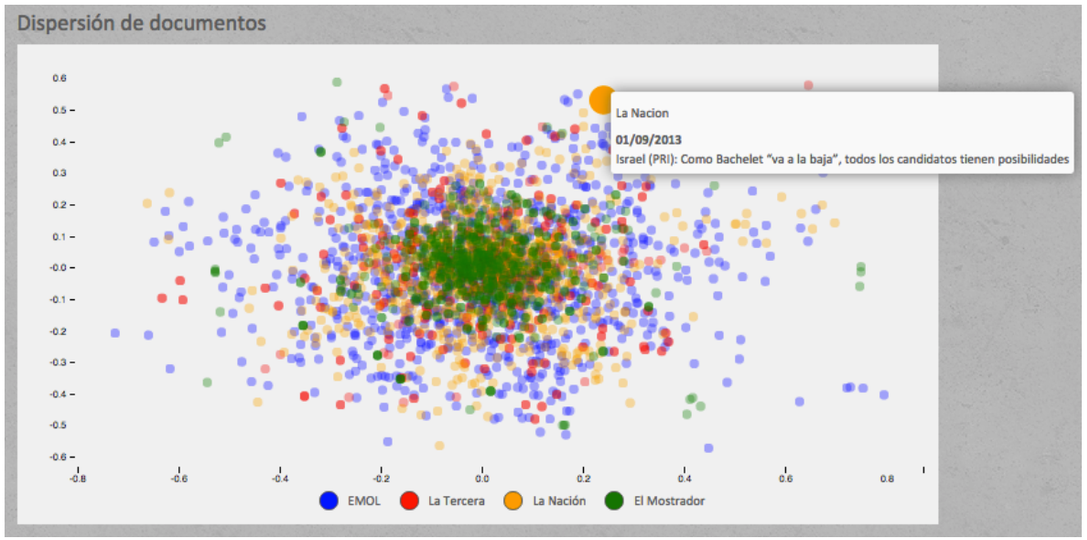
# LSI IV

| | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ |
|---|---|---|---|---|---|---|
| 1 | −0.75 | −0.28 | −0.20 | −0.45 | −0.33 | −0.12 |
| 2 | −0.29 | −0.53 | −0.19 | 0.63 | 0.22 | 0.41 |
| 3 | 0.28 | −0.75 | 0.45 | −0.20 | 0.12 | −0.33 |
| 4 | 0.00 | 0.00 | 0.58 | 0.00 | −0.58 | 0.58 |
| 5 | −0.53 | 0.29 | 0.63 | 0.19 | 0.41 | −0.22 |



# LSI IV

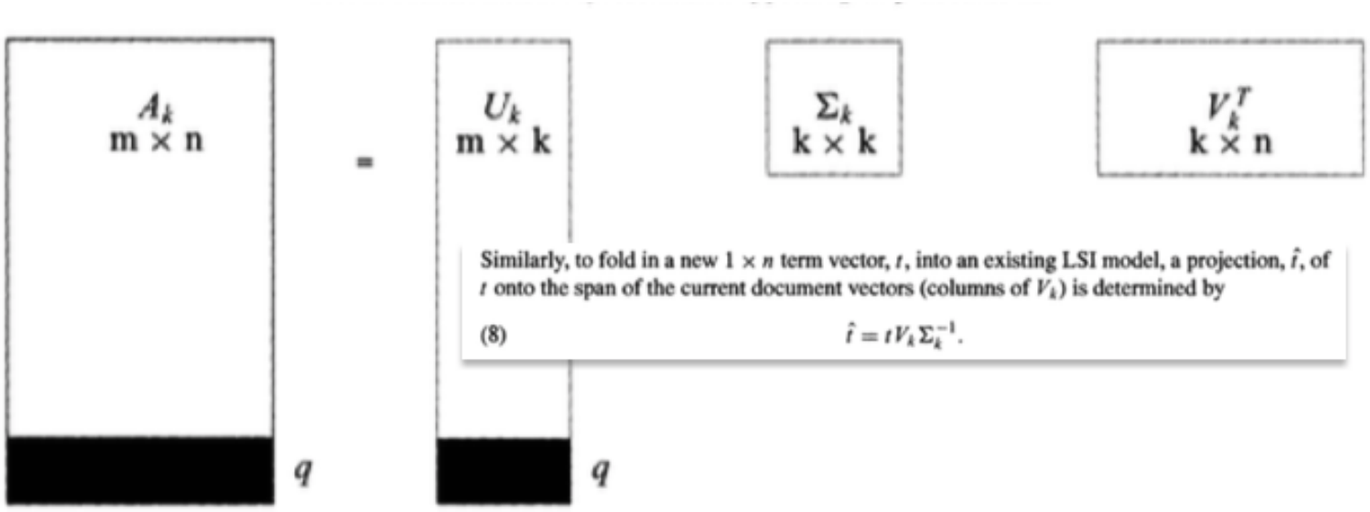
Demo: http://dfao-uc.github.io/




21/28

# Proyección de documentos o términos nuevos

- Folding in: Using Linear Algebra for Intelligent Information Retrieval




Similarly, to fold in a new $1 \times n$ term vector, $t$, into an existing LSI model, a projection, $\hat{t}$, of $t$ onto the span of the current document vectors (columns of $V_k$) is determined by

$$\hat{t} = t V_k \Sigma_k^{-1}. \quad (8)$$

FIG. 3. *Mathematical representation of folding-in q terms.*

# LDA I

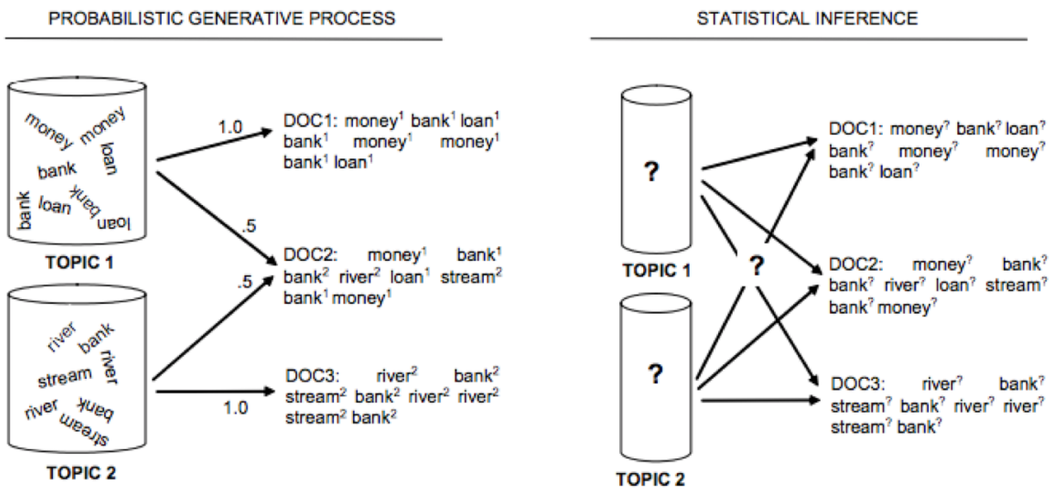
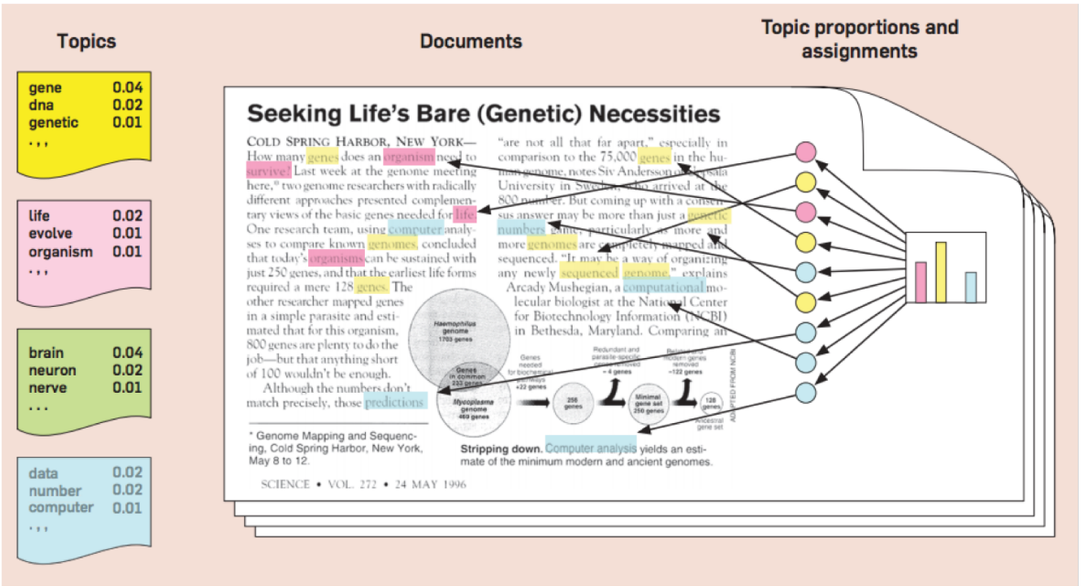


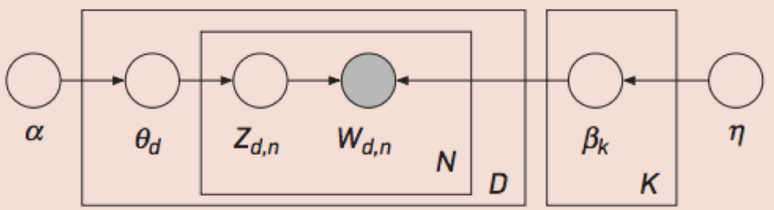

**Figure 2.** Illustration of the generative process and the problem of statistical inference underlying topic models

# LDA II


Topics
gene 0.04
dna 0.02
genetic 0.01
life 0.02
evolve 0.01
organism 0.01
brain 0.04
neuron 0.02
nerve 0.01
data 0.02
number 0.02
computer 0.01
Documents
Topic proportions and assignments
Seeking Life's Bare (Genetic) Necessities
COLD SPRING HARBOR, NEW YORK—
How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.
Although the numbers don't match precisely, those predictions
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a puzzle. Such numbers games, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
SCIENCE • VOL. 272 • 24 MAY 1996

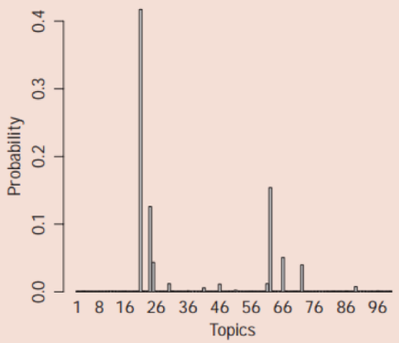
# LDA III

**Figure 4. The graphical model for latent Dirichlet allocation. Each node is a random variable and is labeled according to its role in the generative process (see Figure 1). The hidden nodes—the topic proportions, assignments, and topics—are unshaded. The observed nodes—the words of the documents—are shaded. The rectangles are "plate" notation, which denotes replication. The *N* plate denotes the collection words within documents; the *D* plate denotes the collection of documents within the collection.**



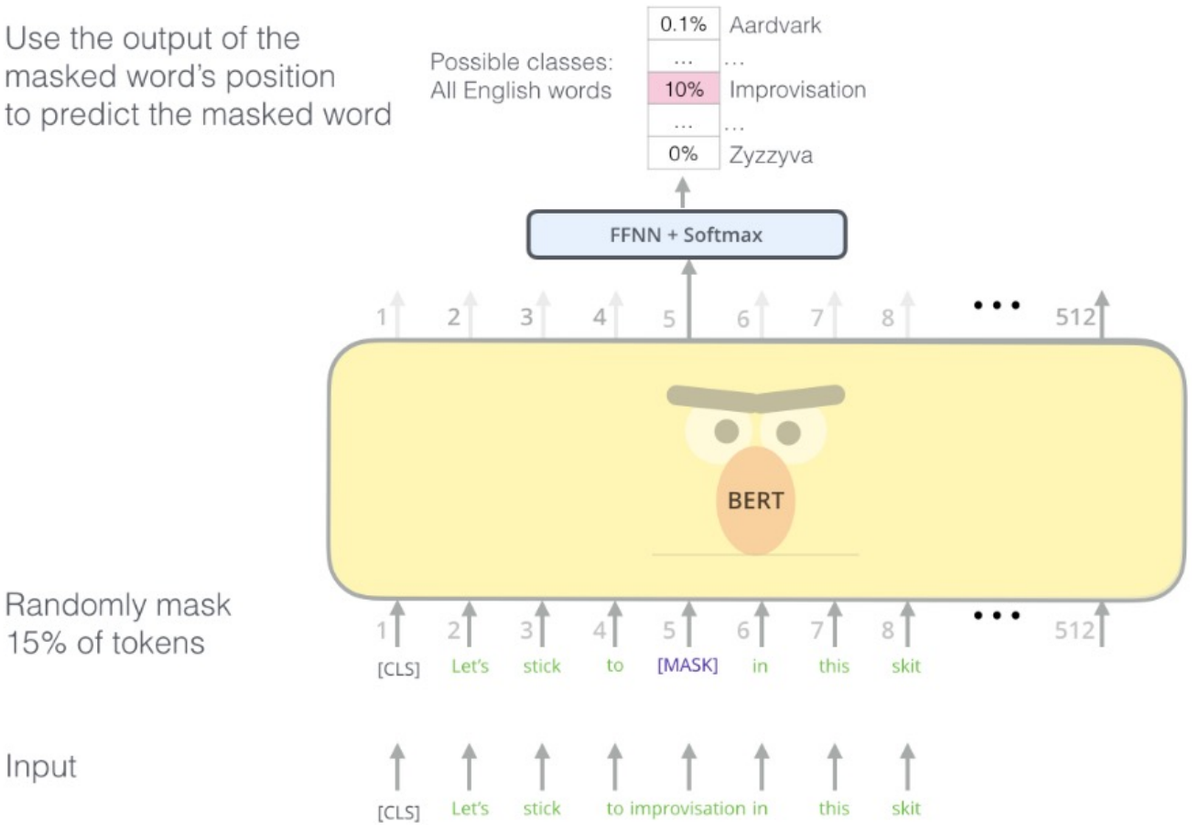

26/28

# LDA IV




| "Genetics" | "Evolution" | "Disease" | "Computers" |
|---|---|---|---|
| human | evolution | disease | computer |
| genome | evolutionary | host | models |
| dna | species | bacteria | information |
| genetic | organisms | diseases | data |
| genes | life | resistance | computers |
| sequence | origin | bacterial | system |
| gene | biology | new | network |
| molecular | groups | strains | systems |
| sequencing | phylogenetic | control | model |
| map | living | infectious | parallel |
| information | diversity | malaria | methods |
| genetics | group | parasite | networks |
| mapping | new | parasites | software |
| project | two | united | new |
| sequences | common | tuberculosis | simulations |

# BERT




BERT's clever language modeling task masks 15% of words in the input and asks the model to predict the missing word.

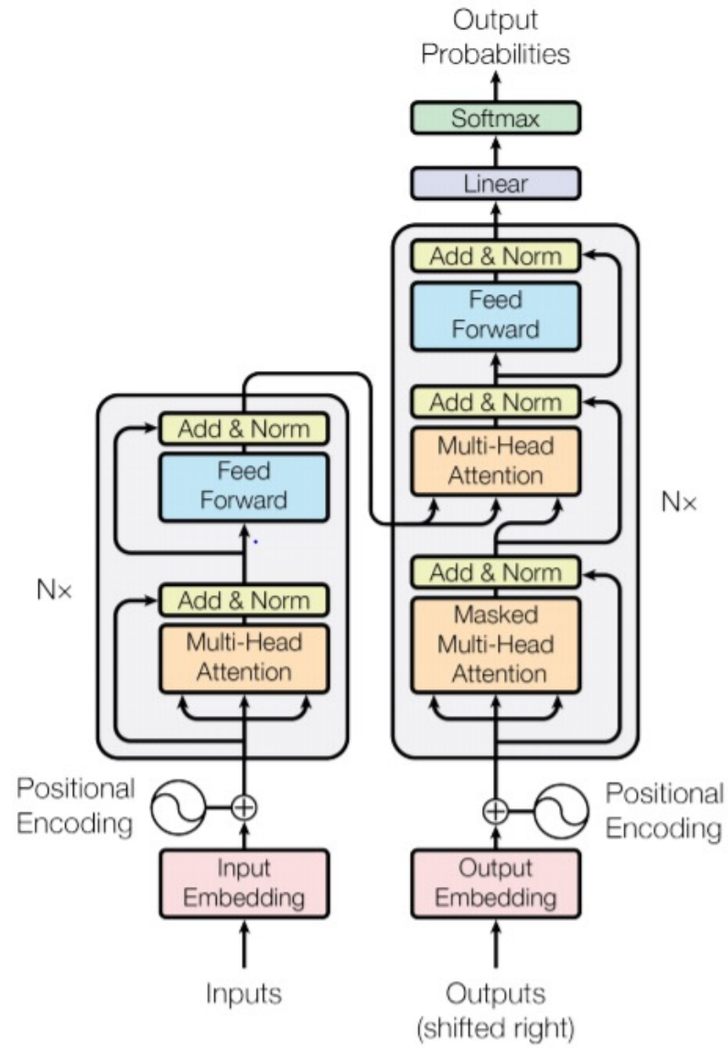
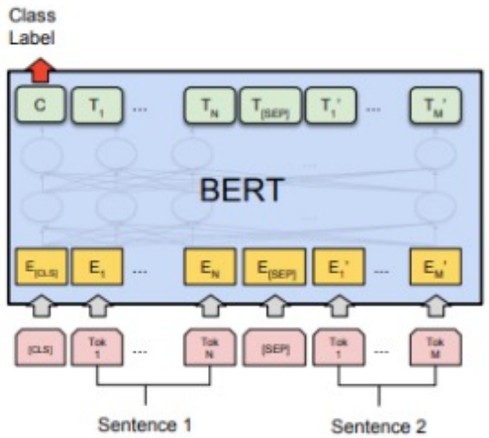
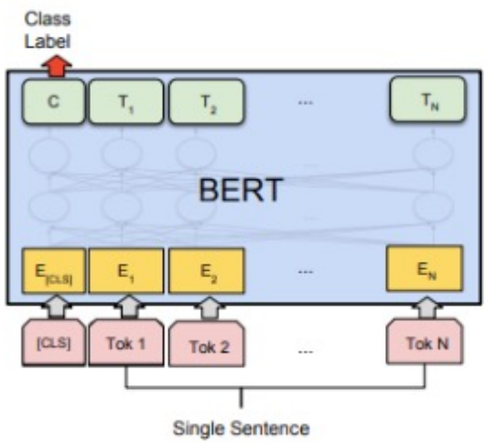



Figure 1: The Transformer - model architecture.

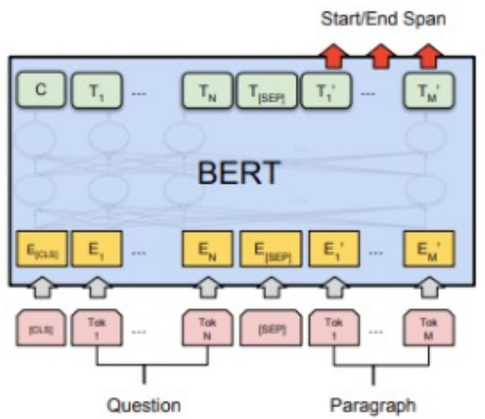
# BERT for different tasks



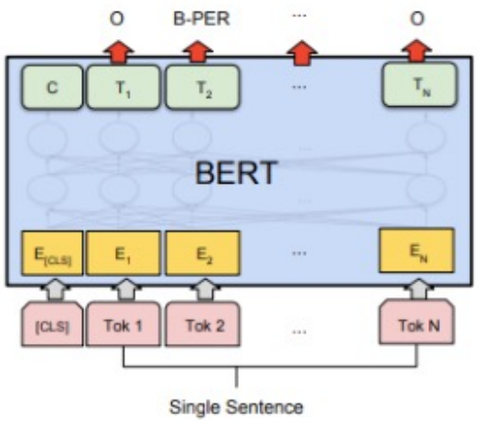

(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG




(b) Single Sentence Classification Tasks: SST-2, CoLA




(c) Question Answering Tasks: SQuAD v1.1

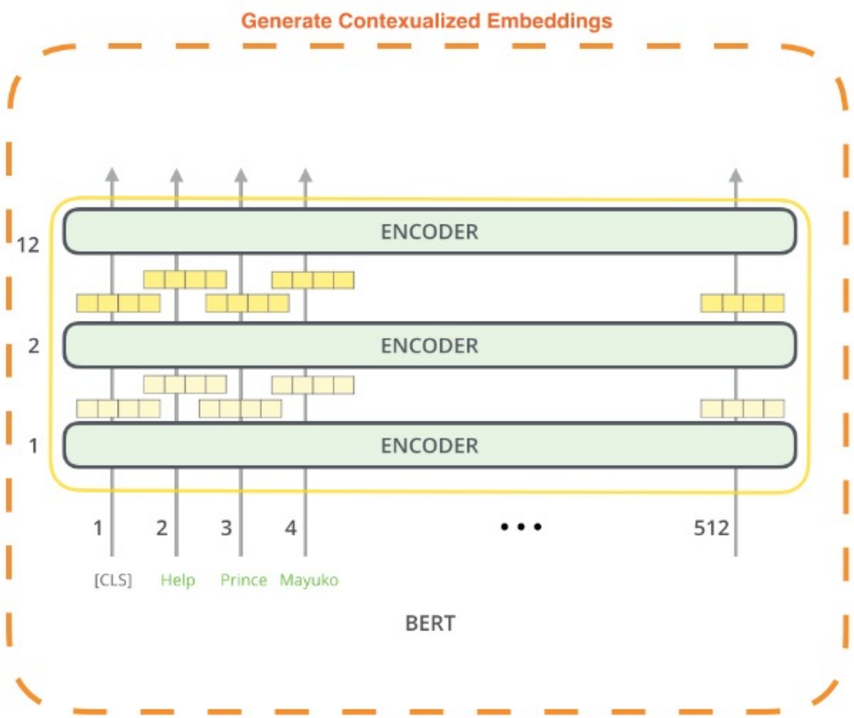



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

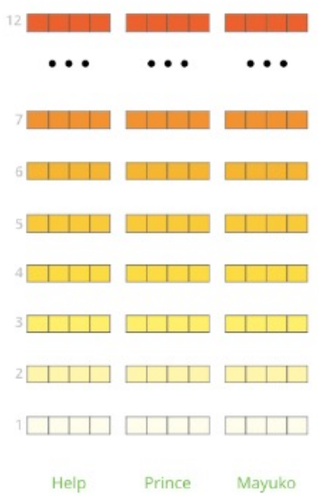
# BERT for feature extraction

**BERT for feature extraction**

The fine-tuning approach isn't the only way to use BERT. Just like ELMo, you can use the pre-trained BERT to create contextualized word embeddings. Then you can feed these embeddings to your existing model – a process the paper shows yield results not far behind fine-tuning BERT on a task such as named-entity recognition.




The output of each encoder layer along each token's path can be used as a feature representing that token.




**But which one should we use?**

# Neural Networks for text-aware content-based recommendation

**Content-Based Citation Recommendation**

**Chandra Bhagavatula**
Allen Institute for AI
chandrab@allenai.org

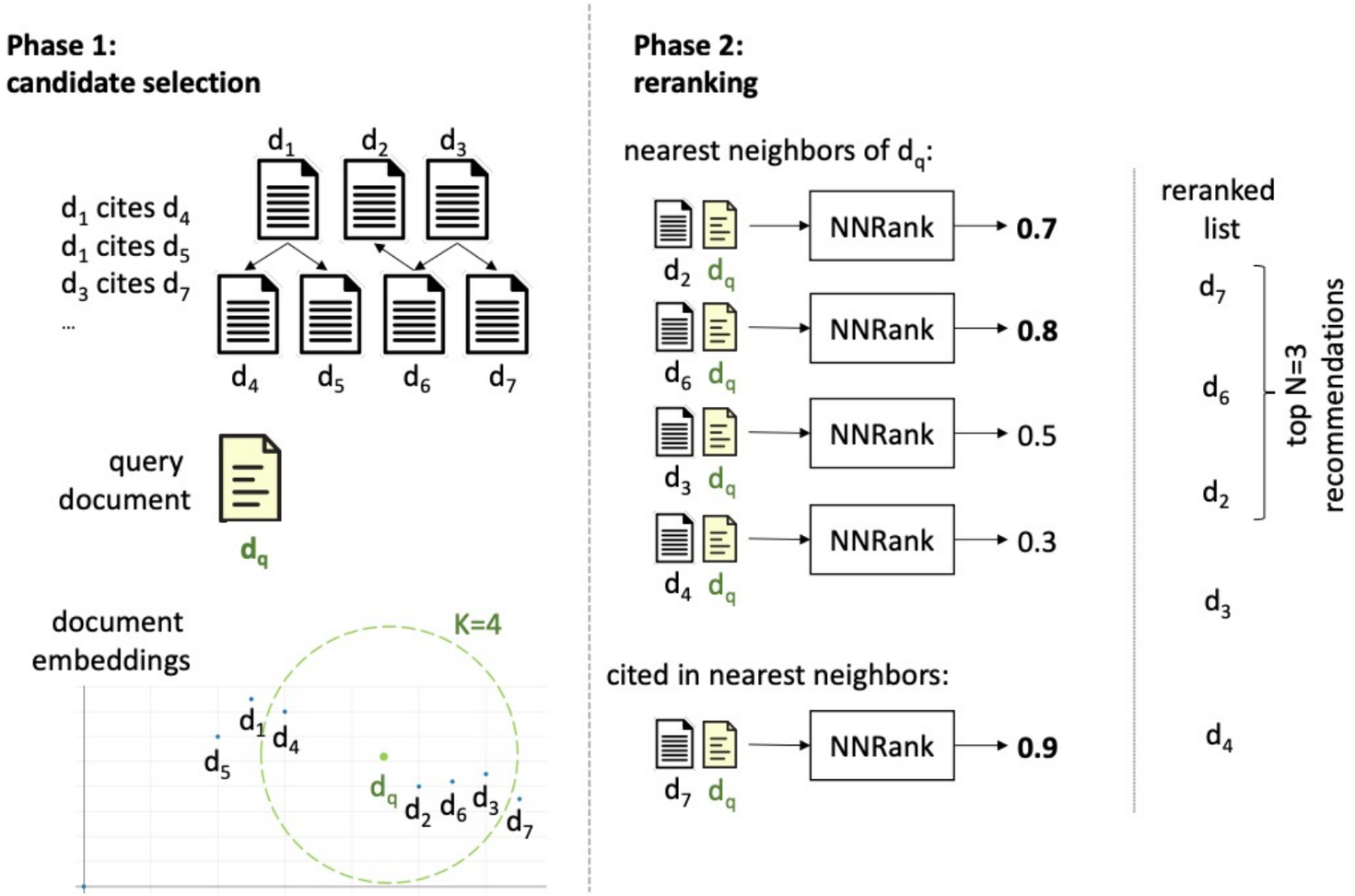
**Sergey Feldman**
Data Cowboys *
sergey@data-cowboys.com

**Russell Power**
Independent Researcher †
russell.power@gmail.com

**Waleed Ammar**
Allen Institute for AI
waleeda@allenai.org

https://aclanthology.org/N18-1022.pdf

# 2-stage model: filter and re-ranking



# NNRank of stage 2

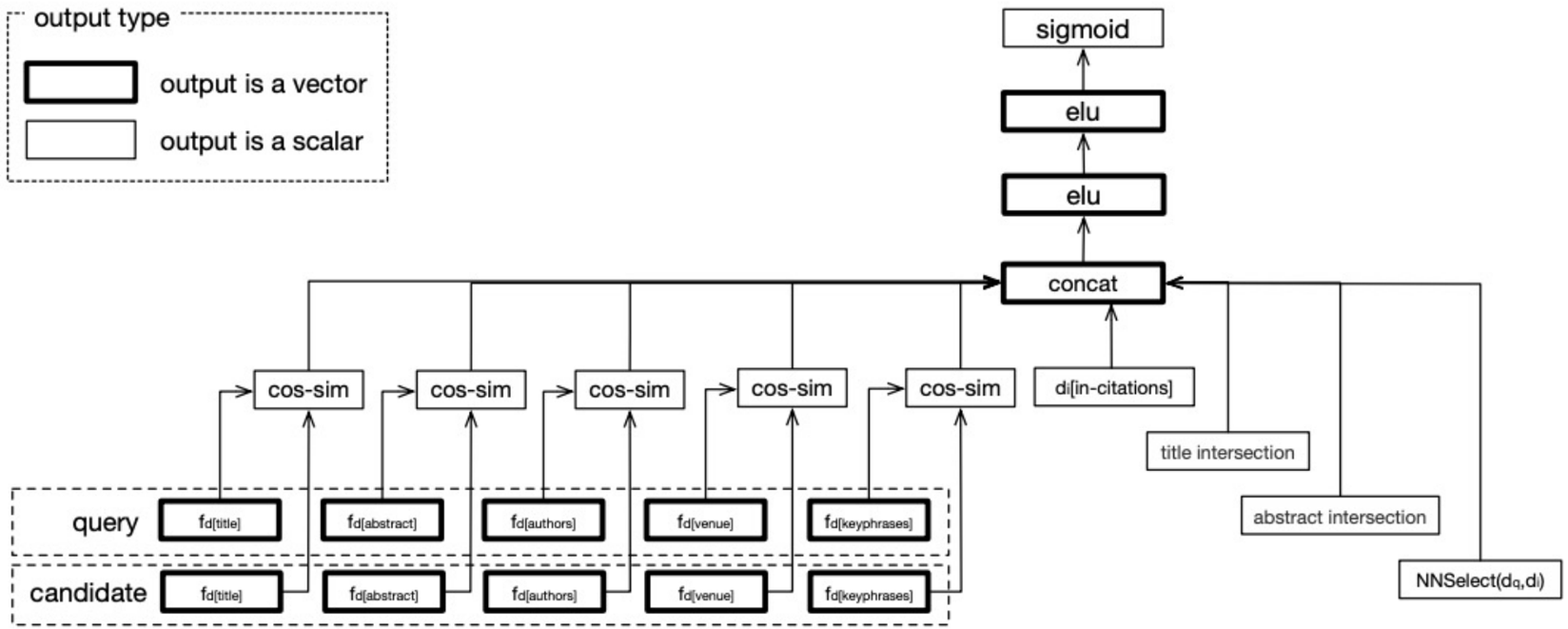



Figure 2: `NNRank` architecture. For each of the textual and categorical fields, we compute the cosine similarity between the embedding for $d_q$ and the corresponding embedding for $d_i$. Then, we concatenate the cosine similarity scores, the numeric features and the summed weights of the intersection words, followed by two dense layers with ELU non-linearities The output layer is a dense layer with sigmoid non-linearity, which estimates the probability that $d_q$ cites $d_i$.

# Results

| Method | DBLP | | PubMed | | OpenCorpus | |
|---|---|---|---|---|---|---|
| | F1@20 | MRR | F1@20 | MRR | F1@20 | MRR |
| BM25 | 0.119 | 0.425 | 0.209 | 0.574 | 0.058 | 0.218 |
| ClusCite | 0.237 | 0.548 | 0.274 | 0.578 | – | – |
| NNSelect | 0.282±0.002 | 0.579±0.007 | 0.309±0.001 | 0.699±0.001 | 0.109 | 0.221 |
| + NNRank | 0.302±0.001 | 0.672±0.015 | 0.325±0.001 | 0.754±0.003 | **0.126** | 0.330 |
| + metadata | **0.303**±0.001 | **0.689**±0.011 | **0.329**±0.001 | **0.771**±0.003 | 0.125 | **0.330** |

Table 2: F1@20 and MRR results for two baselines and three variants of our method. BM25 results are based on our implementation of this baseline, while ClusCite results are based on the results reported in Ren et al. (2014). "NNSelect" ranks candidates using cosine similarity between the query and candidate documents in the embedding space (phase 1). "NNSelect + NNRank" uses the discriminative reranking model to rerank candidates (phase 2), without encoding any of the metadata features. "+ metadata" encodes the metadata features (i.e., keyphrases, venues and authors), achieving the best results on all datasets. Mean and standard deviations are reported based on five trials.

# News recommendation

**UNBERT: User-News Matching BERT for News Recommendation**

**Qi Zhang , Jingjie Li* , Qinglin Jia , Chuyuan Wang , Jieming Zhu , Zhaowei Wang and Xiuqiang He**

Huawei Noah's Ark Lab

{zhangqi193, lijingjie1, jiaqinglin2, wangchuyuan, jamie.zhu, wangzhaowei3, hexiuqiang1}@huawei.com

https://www.ijcai.org/proceedings/2021/0462.pdf

# Objetivo de UNBERT

- Separar la contribución a nivel de palabra y a nivel de noticia

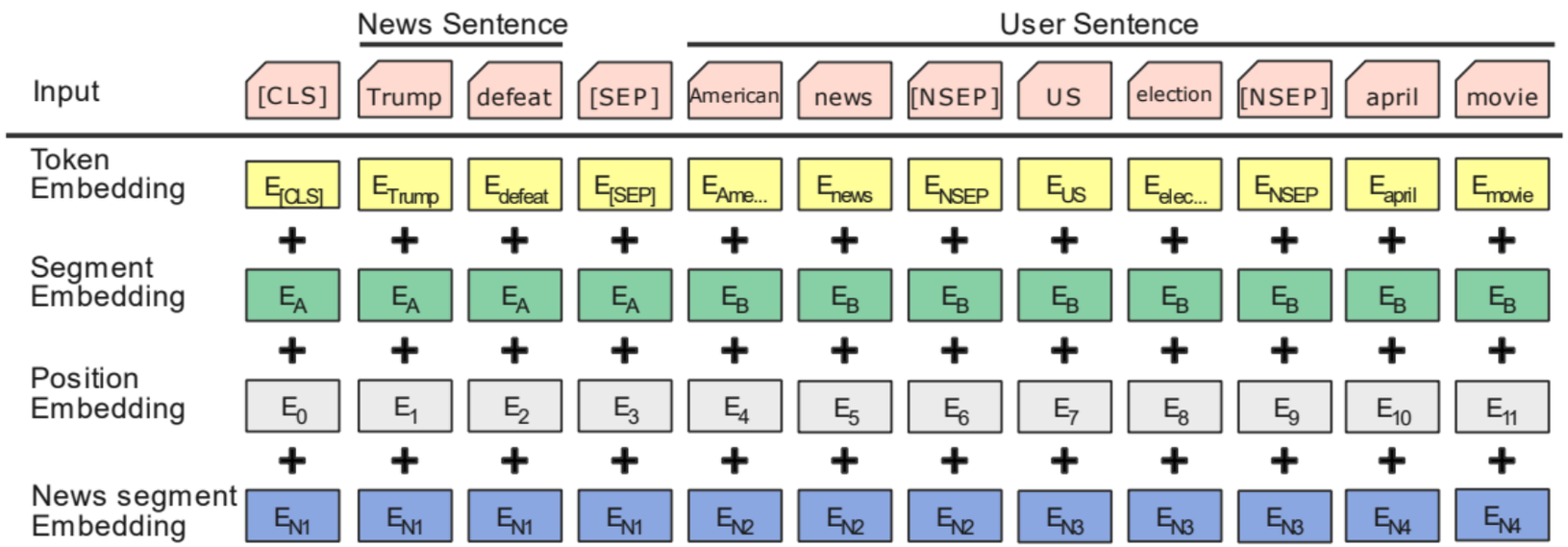



Figure 2: UNBERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings, the position embeddings, and the News segmentation embeddings.

# Arquitectura de UNBERT

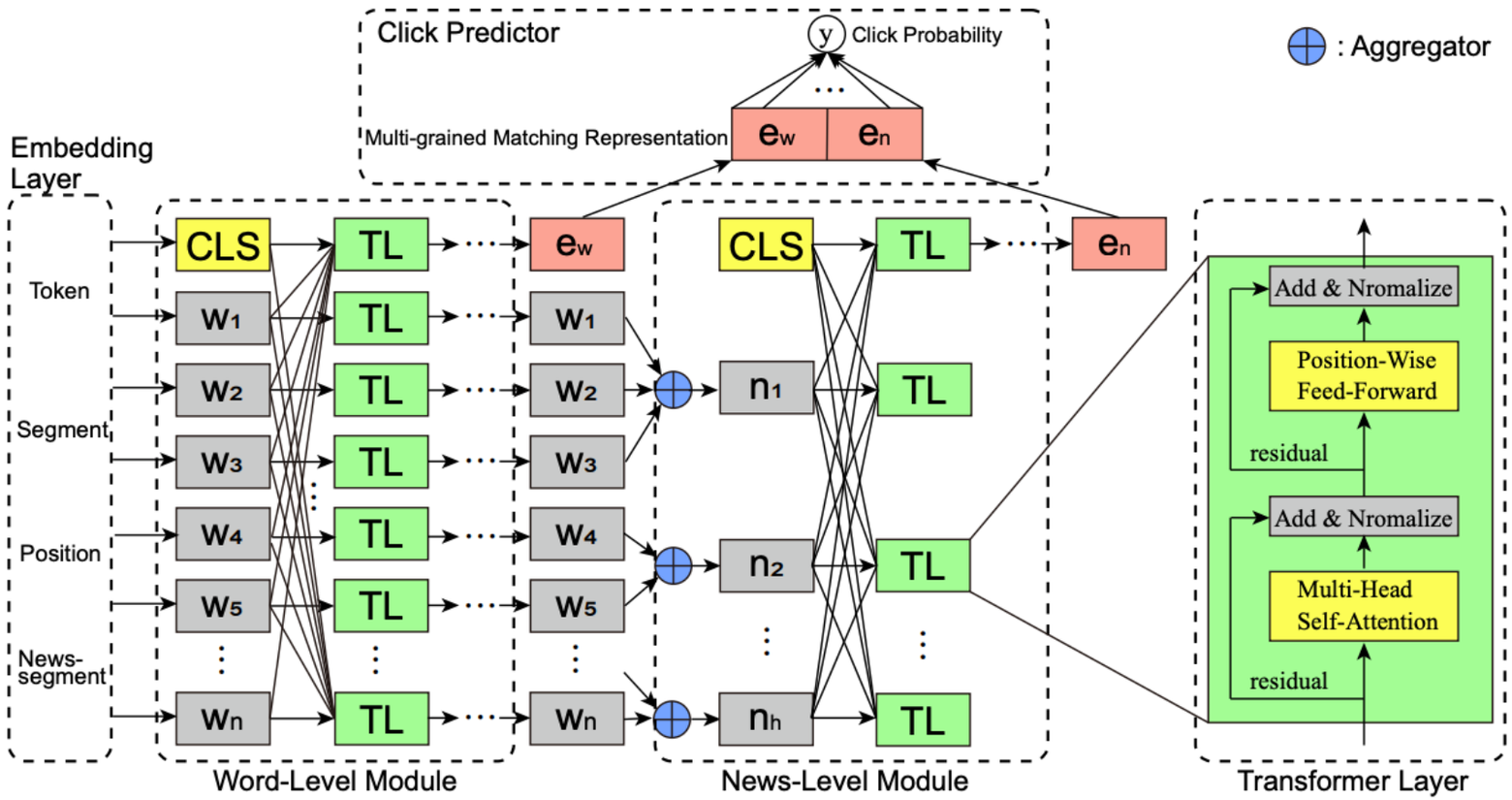



Figure 3: The overall architecture of our UNBERT approach.

# Resultados de UNBERT de UNBERT

| Method | MIND-small | | | | MIND-large | | | |
|---|---|---|---|---|---|---|---|---|
| | AUC | MRR | nDCG@5 | nDCG@10 | AUC | MRR | nDCG@5 | nDCG@10 |
| LibFM | 0.5974 | 0.2633 | 0.2795 | 0.3429 | 0.6185 | 0.2945 | 0.3145 | 0.3713 |
| DeepFM | 0.5989 | 0.2621 | 0.2774 | 0.3406 | 0.6187 | 0.2930 | 0.3135 | 0.3705 |
| DKN | 0.6175 | 0.2705 | 0.2890 | 0.3538 | 0.6407 | 0.3042 | 0.3292 | 0.3866 |
| NPA | 0.6321 | 0.2911 | 0.3170 | 0.3781 | 0.6592 | 0.3207 | 0.3472 | 0.4037 |
| NAML | 0.6550 | 0.3039 | 0.3308 | 0.3931 | 0.6646 | 0.3275 | 0.3566 | 0.4140 |
| LSTUR | 0.6438 | 0.2946 | 0.3189 | 0.3817 | 0.6708 | 0.3236 | 0.3515 | 0.4093 |
| NRMS | 0.6483 | 0.3001 | 0.3252 | 0.3892 | 0.6766 | 0.3325 | 0.3628 | 0.4198 |
| FIM | 0.6502 | 0.3026 | 0.3291 | 0.3910 | 0.6787 | 0.3346 | 0.3653 | 0.4221 |
| UNBERT | **0.6762** | **0.3172** | **0.3475** | **0.4102** | **0.7068** | **0.3568** | **0.3913** | **0.4478** |
| %Improv. | 2.12 | 1.33 | 1.67 | 1.71 | 2.81 | 2.22 | 2.60 | 2.57 |
| UNBERT-en$^{\triangle}$ | - | - | - | - | 0.7183 | 0.3659 | 0.4020 | 0.4581 |

Boldface indicates the best results (the higher, the better), while the second best is underlined. UNBERT-en$^{\triangle}$ represents the ensemble score based on UNBERT which is at the top of https://msnews.github.io/#leaderboard.

# Referencias


- Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval (Vol. 1, p. 6). Cambridge: Cambridge university press.
- Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. Handbook of latent semantic analysis, 427(7), 424-440.
- Blei, D. M. (2012). Probabilistic topic models. Communications of the ACM, 55(4), 77-84.