



TDC 2019

Gabriel Moreira  
Lead Data Scientist



# Deep Learning for Recommender Systems



# About me

**Gabriel Moreira**



@gspmoreira





DRIVEN BY  
**IMPACT**

We are **digital transformation agents** for the most valuable brands in the world, generating **business impact** for all projects we lead.



# CI&T Cognitive Solutions

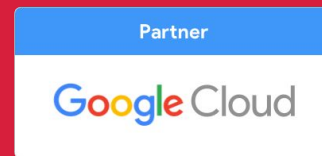


Investing in Machine Learning **since 2012**



## Recognized Expertise

Google ML Specialized Partner  
Tensorflow.org Reference



## End-to-End

Machine Learning  
Capabilities

Life is too short!



*"We are leaving the  
Information Age and  
entering the  
Recommendation Age."  
Cris Anderson, "The long tail"*



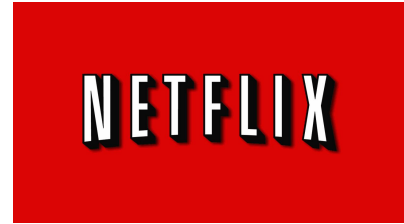
Recommendations are responsible for...

**amazon.com**<sup>®</sup>

38% of sales

**Google** news

39% of top news  
visualization



75% of watched content

# What else may I recommend?

products  
tags  
professionals  
courses  
musics movies  
jobs books  
papers girlfriends  
investments restaurants  
videos  
dressing





# What can a Recommender Systems do?

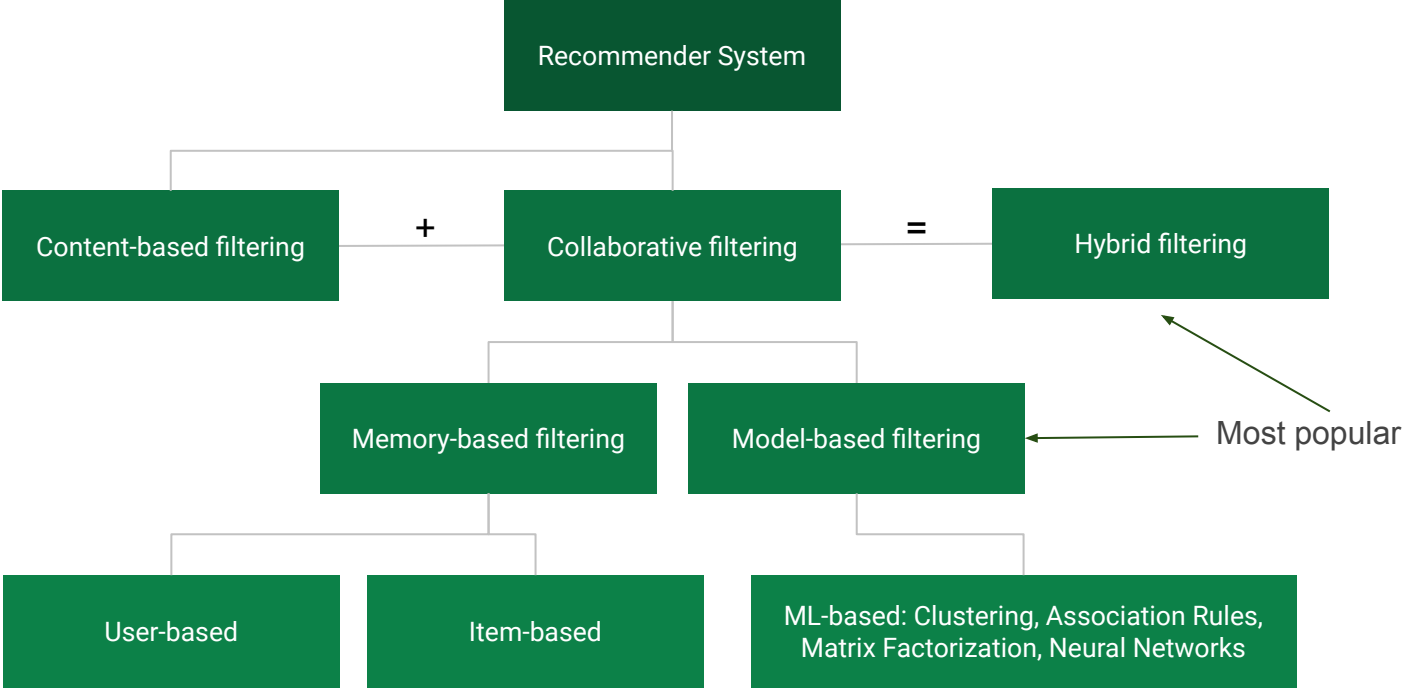
## **1 - Recommendation**

Given a user, produce an ordered list matching the user needs

## **2 - Prediction**

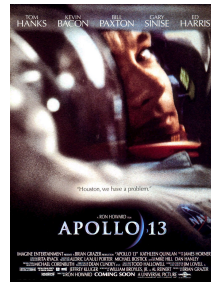
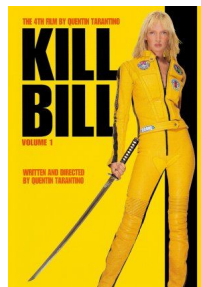
Given an item, what is its relevance for each user?

# Recommender System Methods



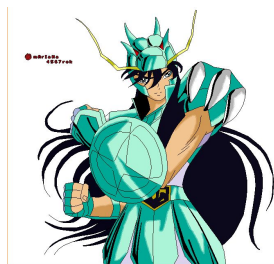
# Collaborative Filtering

# User-Based Collaborative Filtering



Likes

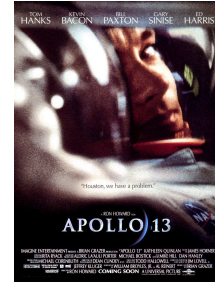
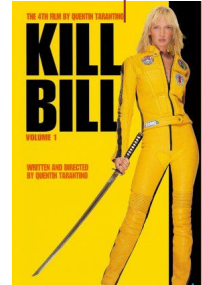
Recommends



Similar interests

# Item-Based Collaborative Filtering

Who likes A also likes B



Likes



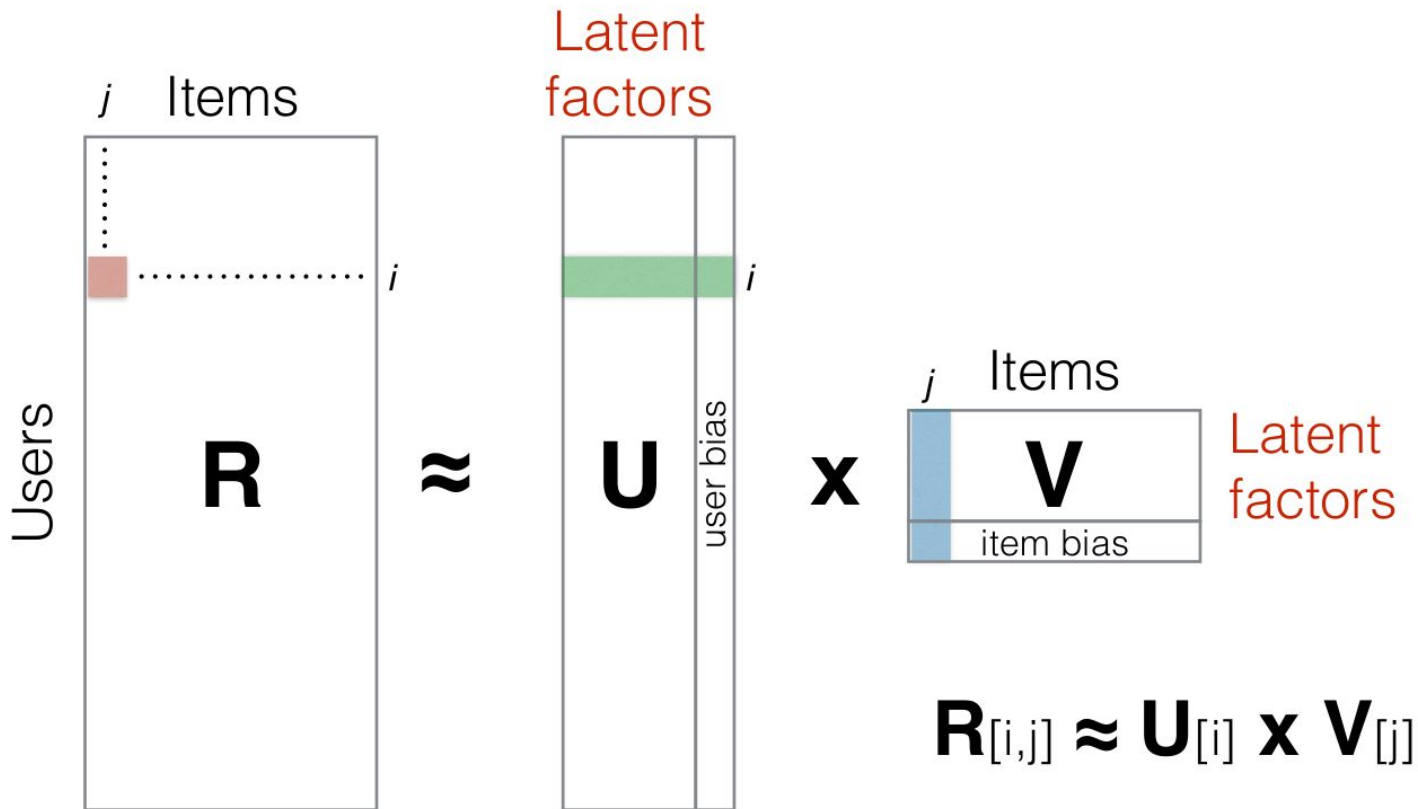
Likes

Likes



Recommends

# Collaborative Filtering based on Matrix Factorization



# Collaborative Filtering

## Advantages

- Works to any item kind (ignore attributes)

## Drawbacks

- Usually recommends more popular items
- Cold-start
  - Cannot recommend items not already rated/consumed
  - Needs a minimum amount of users to match similar users

# Frameworks - Recommender Systems



Java



Python / Scala



Python



Java



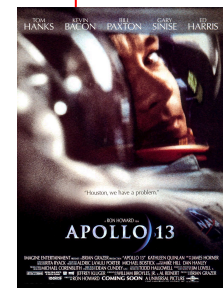
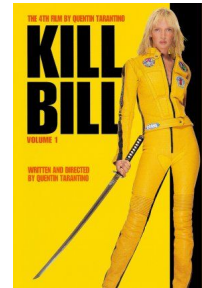
.NET



# Content-Based Filtering

# Content-Based Filtering

Similar content (e.g. actor)



Likes



Recommends

# Content-Based Filtering

## Advantages

- Does not depend upon other users
- May recommend new and unpopular items
- Recommendations can be easily explained

## Drawbacks

- Overspecialization
- May be difficult to extract attributes from audio, movies or images

# Hybrid Recommender Systems

## Some approaches...

### **Composite**

Iterates by a chain of algorithm, aggregating recommendations.

### **Weighted**

Each algorithm has as a weight and the final recommendations are defined by weighted averages.



A diver is seen from above, swimming in a dark, narrow underwater cave. The water is a deep, dark blue-green color. The cave walls are rocky and textured. The diver is positioned in the center of the frame, with their head pointing towards the bottom right. The lighting is dramatic, with a bright light source from above creating a shimmering effect on the water's surface.

# Deep Recommender Systems

# Why Deep Learning has a potential for RecSys?

## 1. Feature extraction directly from the content (e.g., image, text, audio)



**Images**

- CNN



**Text**


- CNN
- RNNs
- Weighted word embeddings



**Audio/Music**

- CNN
- RNN

# Why Deep Learning has a potential for RecSys?

- 
2. **Heterogeneous data** handled easily
  3. **Dynamic behaviour** modeling with **RNNs**
  4. More accurate **representation learning of users and items**
    - Natural extensions of CF
  5. RecSys is a complex domain
    - **Deep learning worked well in other complex domains**

# The Deep Learning era of RecSys





# Advances in DL-RecSys

Deep Collaborative Filtering

Learning Item embeddings

Feature Extraction directly from the content

Session-based recommendations with  
RNNs

And their combinations...

---

# News Recommender Systems

# News Recommender Systems

The majority of web traffic (TREVISIOL et al. , 2014b)



The Washington Post

globo.com



YAHOO!  
NEWS

The New York Times



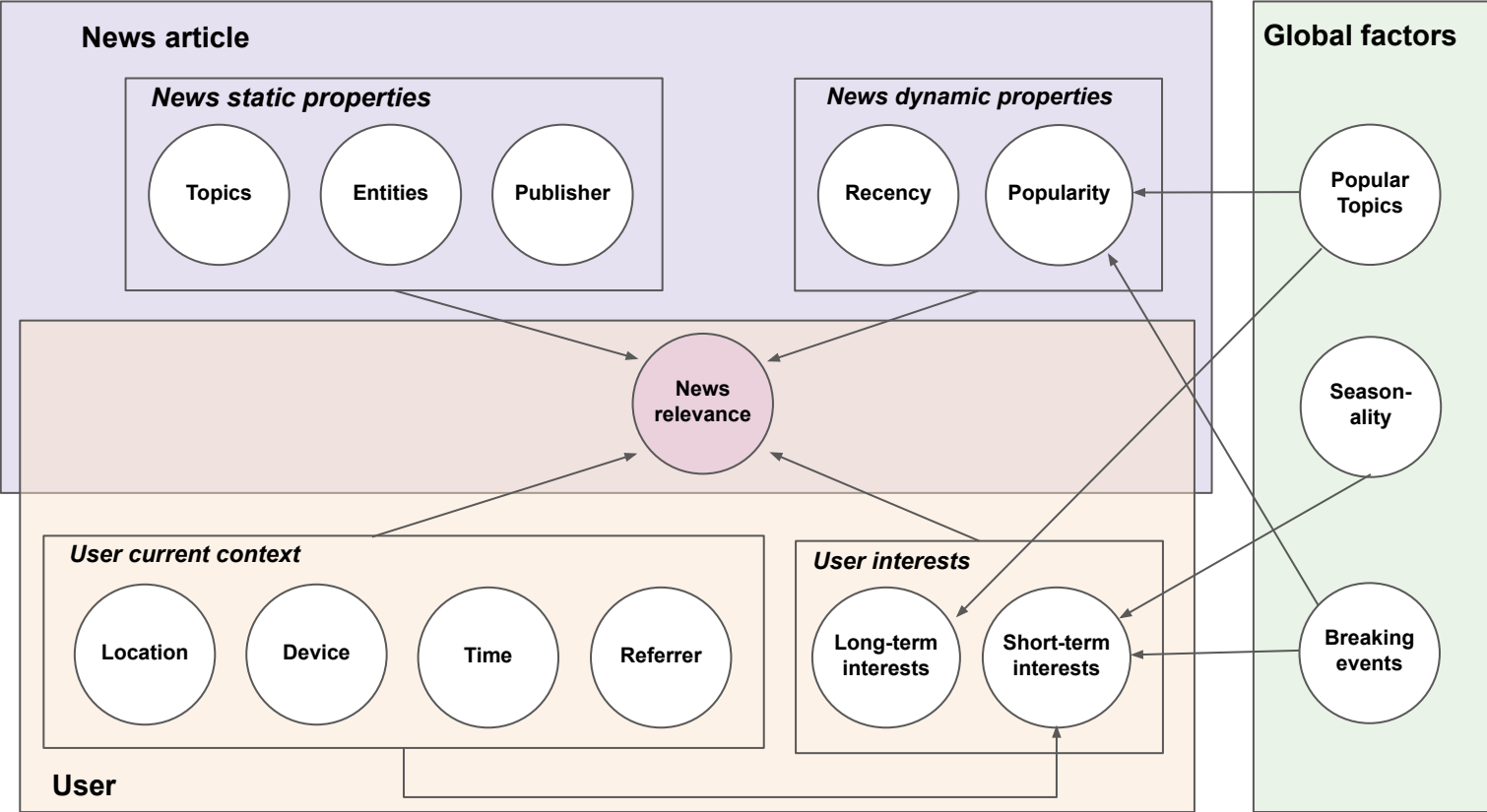
## Challenges

1. Streaming clicks and news articles
2. Most users are anonymous
3. Users' preferences shift
4. Accelerated relevance decay

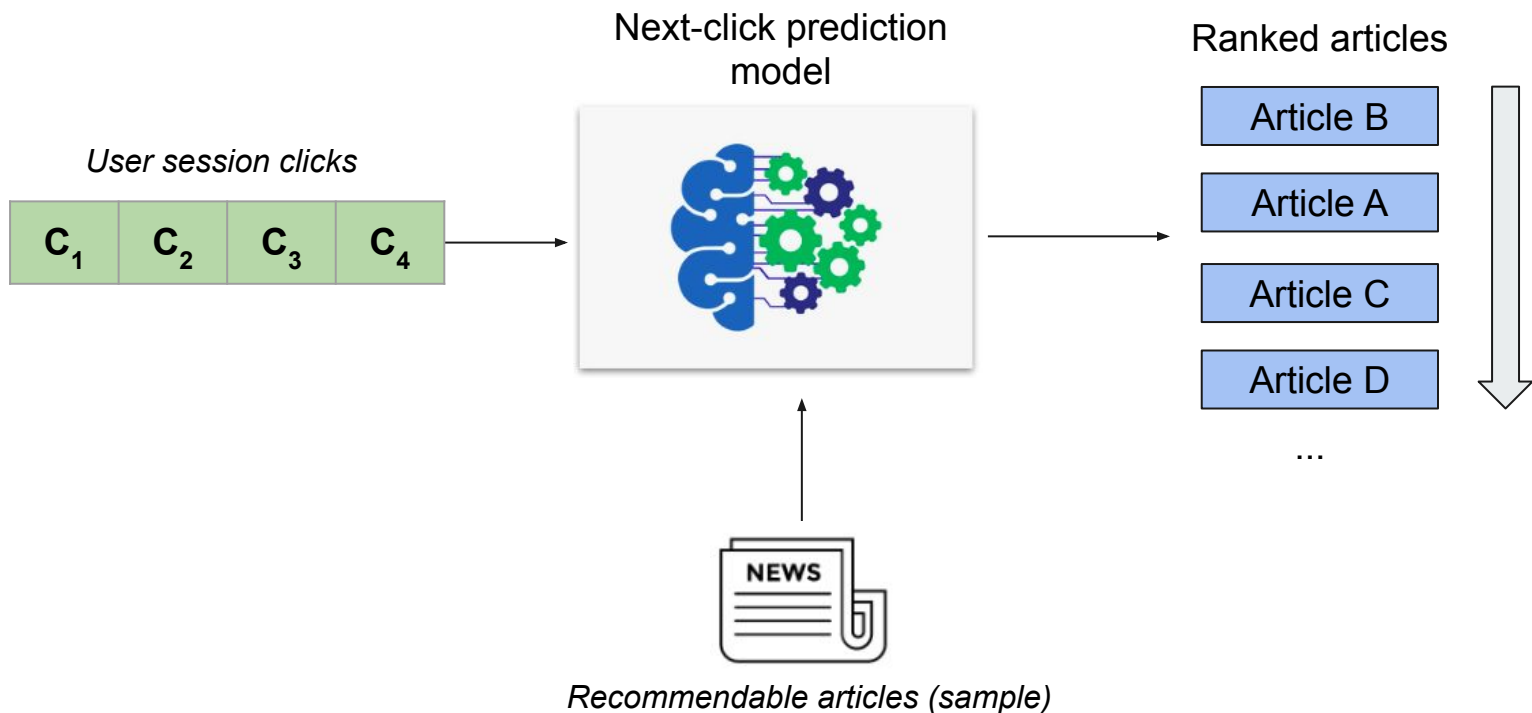
<i>Percentile of clicks</i>	<i>Article age</i>
10%	up to 4 hours
25%	up to 5 hours
50% (Median)	up to 8 hours
75%	up to 14 hours
90%	up to 26 hours

# News Recommender Systems

## Factors affecting news relevance



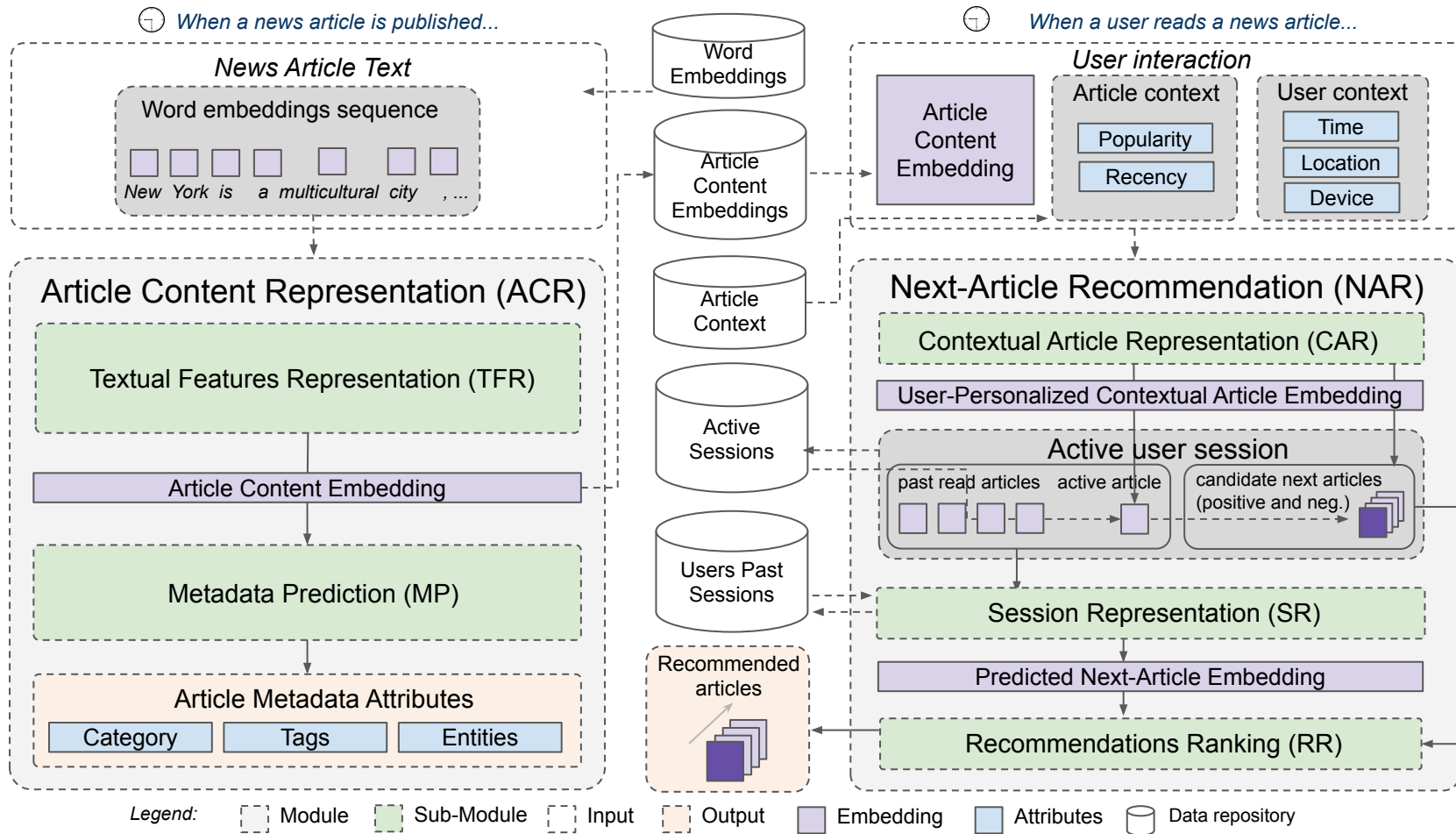
# News session-based recommender overview



---

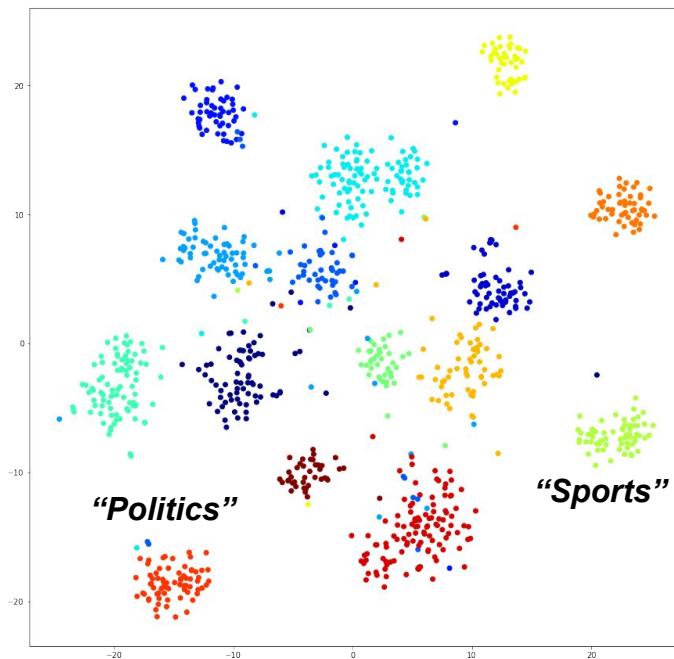
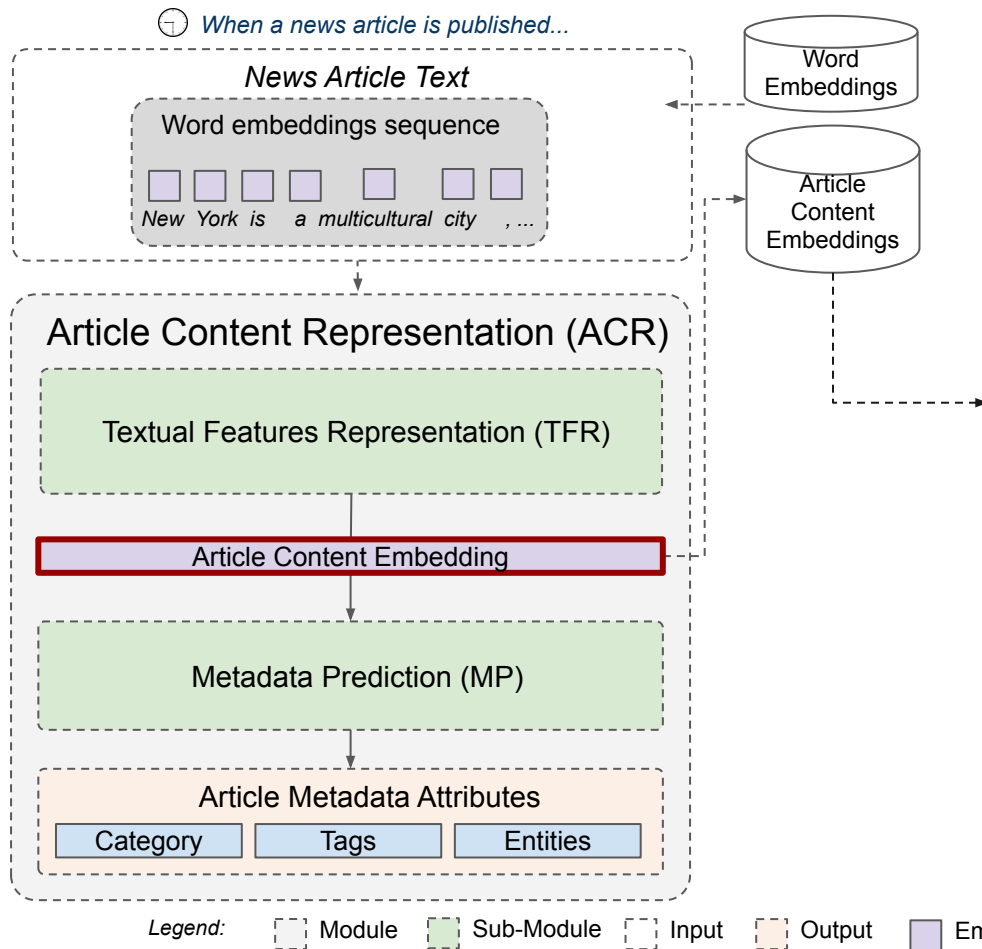
# CHAMELEON: A Deep Learning Meta-Architecture for News Recommendation

# CHAMELEON Meta-Architecture for News RS





# CHAMELEON - ACR module



T-SNE viz. of articles embeddings colored by category, with similar articles highlighted

# CHAMELEON - NAR module

Sessions in a batch

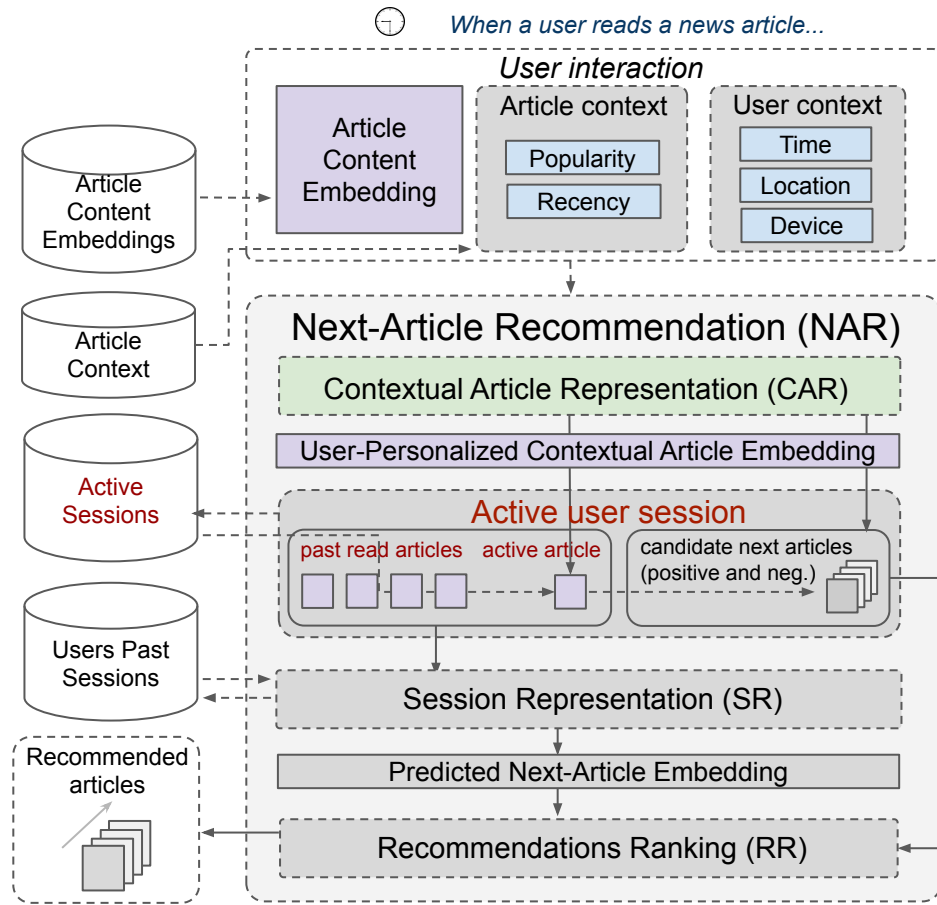
$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	$I_{1,4}$	$I_{1,5}$
$I_{2,1}$	$I_{2,2}$			
$I_{3,1}$	$I_{3,2}$	$I_{3,3}$		

Input

$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	$I_{1,4}$
-----------	-----------	-----------	-----------

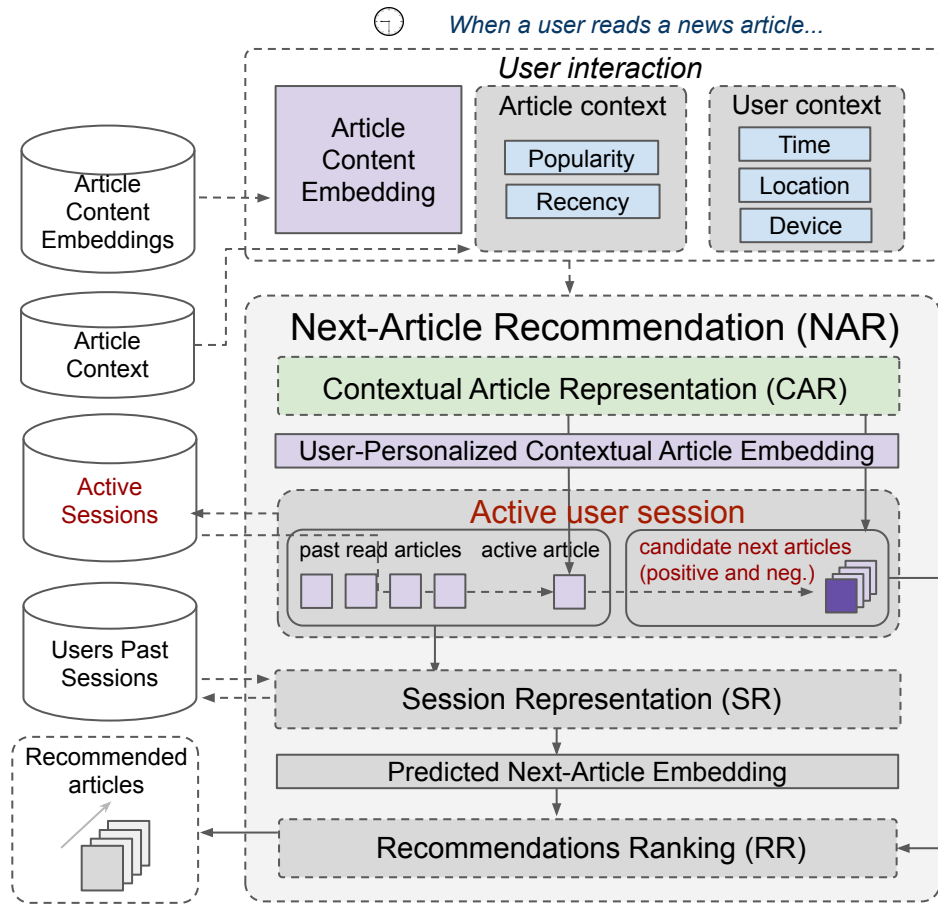
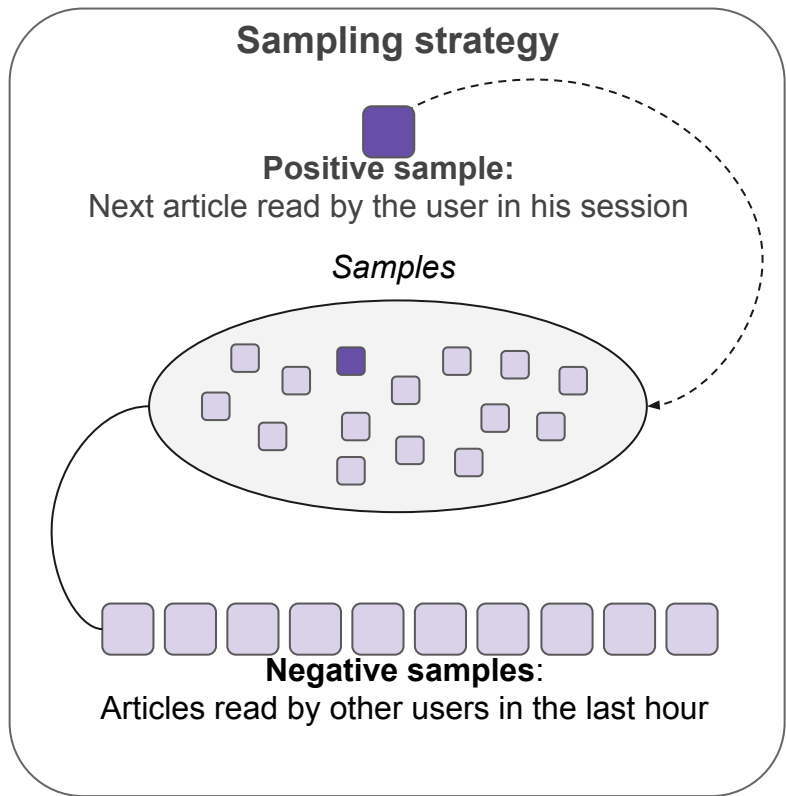
Expected Output (labels)

$I_{1,2}$	$I_{1,3}$	$I_{1,4}$	$I_{1,5}$
-----------	-----------	-----------	-----------



Legend: Module Sub-Module Input Output Embedding Attributes Data repository

# CHAMELEON - NAR module



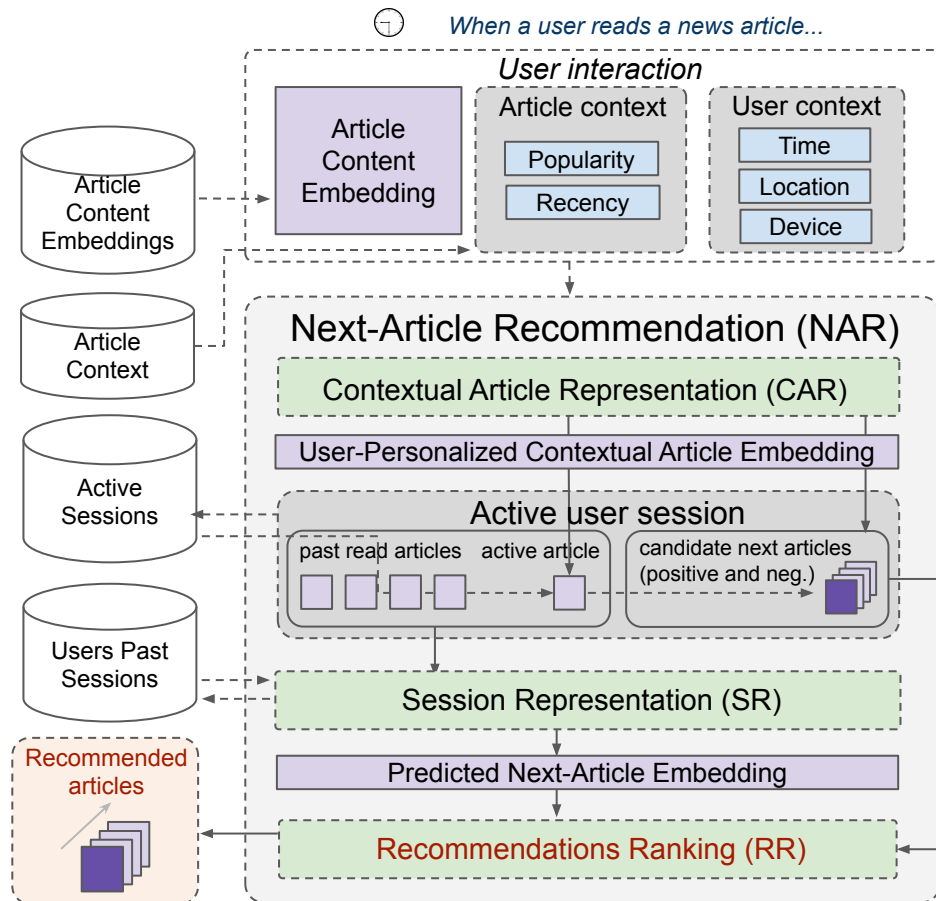
Legend:   Module   Sub-Module   Input   Output   Embedding   Attributes   Data repository

# CHAMELEON - NAR module

## Recommendations Ranking (RR) sub-module

$$R(s, item) = \cos(s, item)$$

Relevance Score of an item for a user session



Legend: Module Sub-Module Input Output Embedding Attributes Data repository

# CHAMELEON - Ranking loss function

$$\cos(\theta) = \frac{a \cdot b}{\|a\| \|b\|}$$

Cosine similarity

$$P(item+ | s) = \frac{\exp(\gamma R(s, item+))}{\sum_{\forall item \in D'} \exp(\gamma R(s, item))}$$

Softmax over Relevance Score

$$l(\theta) = -\log \prod_{(s, item^+)} P(item^+ | s)$$

Loss function

```
#Computing cosine similarities between predicted item embedding and 1 positive | K negative samples
cos_sim_positive = tf.reduce_sum(tf.multiply(tf.nn.l2_normalize(positive_item_embedding, dim=-1),
                                           tf.nn.l2_normalize(predicted_item_embedding, dim=-1)),
                                axis=-1, keep_dims=True)
cos_sim_negative = tf.reduce_sum(tf.multiply(tf.nn.l2_normalize(negative_item_embedding, dim=-1),
                                           tf.nn.l2_normalize(predicted_item_embedding, dim=-1)),
                                axis=-1, keep_dims=True)

#Concatenating cosine similarities
cos_sim_concat_scaled = tf.concat([cos_sim_positive, cos_sim_negative], axis=2) * gamma_var
#Computing softmax over cosine similarities
items_prob = tf.nn.softmax(cos_sim_concat_scaled)
positive_prob = items_prob[:, :, 0]

#Computing loss
loss = -tf.log(positive_prob)
```

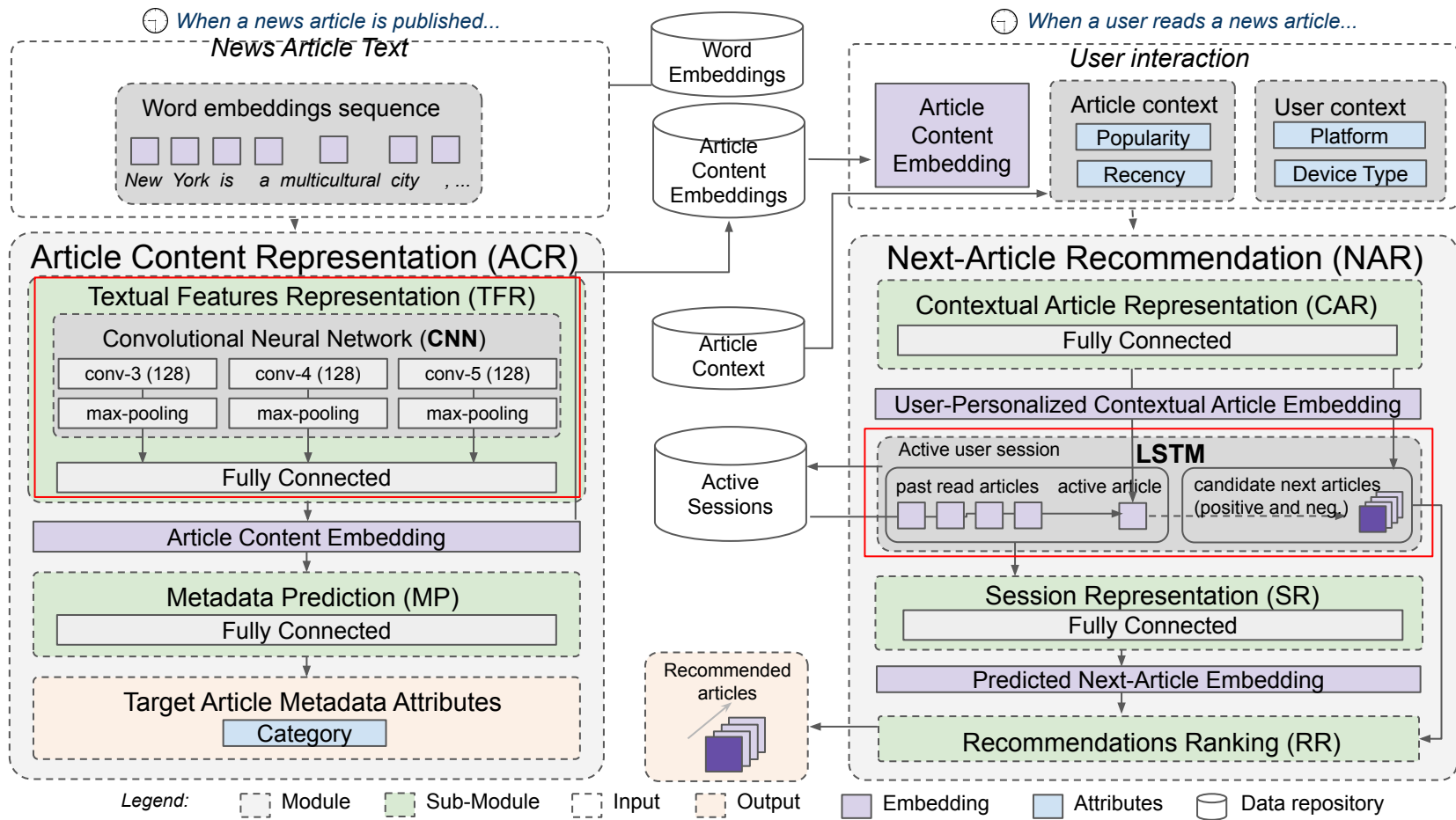
Cosine similarity-based loss function implemented on TensorFlow

---

# CHAMELEON

## Architecture Instatiations

# An architecture instantiation of CHAMELEON (1D CNN and LSTM)



# CHAMELEON Instantiation - Implementation

- CHAMELEON's instantiations are implemented using TensorFlow  
[https://github.com/gabrielspmoreira/chameleon\\_recsys](https://github.com/gabrielspmoreira/chameleon_recsys)

The screenshot shows the GitHub repository page for `gabrielspmoreira / chameleon_recsys`. At the top, there are navigation options: Unwatch (14), Star (56), and Fork (24). Below this is a navigation bar with links for Code, Issues (1), Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings. The main heading is "Source code of CHAMELEON - A Deep Learning Meta-Architecture for News Recommender Systems" with an "Edit" button. A series of topic tags are displayed, including tensorflow, deep-learning, deep-neural-networks, recommender-system, recommendation-system, recommendation-engine, recommendation-algorithms, news-recommendation, lstm-neural-networks, lstm, lstm-neural-network, rnn, rnn-tensorflow, word-embeddings, and word2vec. Below the tags, there is a "Manage topics" section. A summary bar shows 12 commits, 1 branch, 3 releases, 1 contributor, and MIT license. At the bottom, there are buttons for "Branch: master", "New pull request", "Create new file", "Upload files", "Find File", and "Clone or download". A commit message by gabrielspmoreira is visible: "Improvements on ACR module (fixed tokenization, support to GRU based ...".

- Training and evaluation performed in **Google Cloud Platform ML Engine**

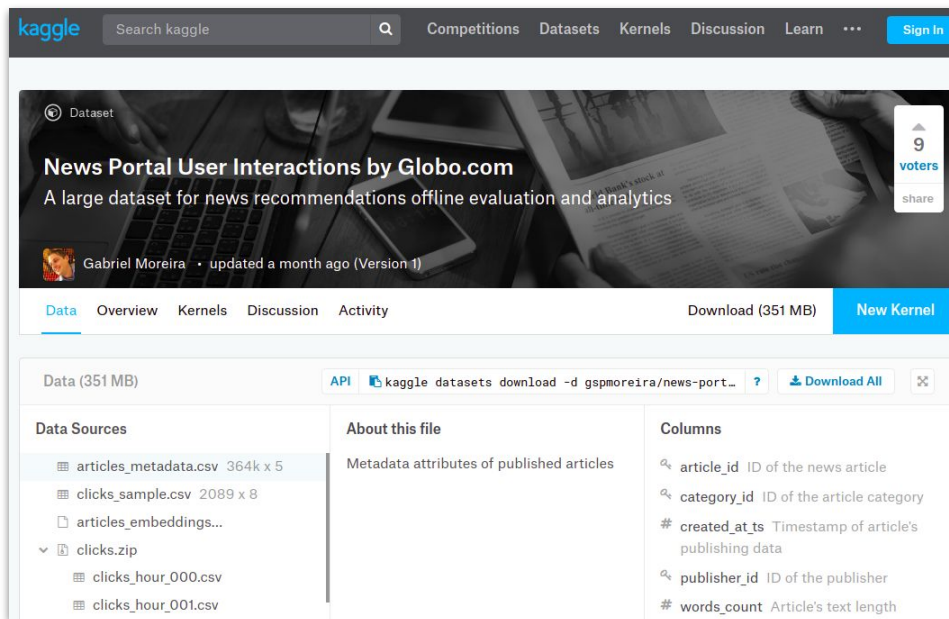




# Experiments

# Experiments - Dataset

- Provided by **Globo.com (G1)**, the most popular news portal in Brazil
- Sample from **Oct., 1 to 16, 2017**, with over **3 M** clicks, distributed in **1.2 M sessions** from **330 K** users, who read over **50 K** unique news articles

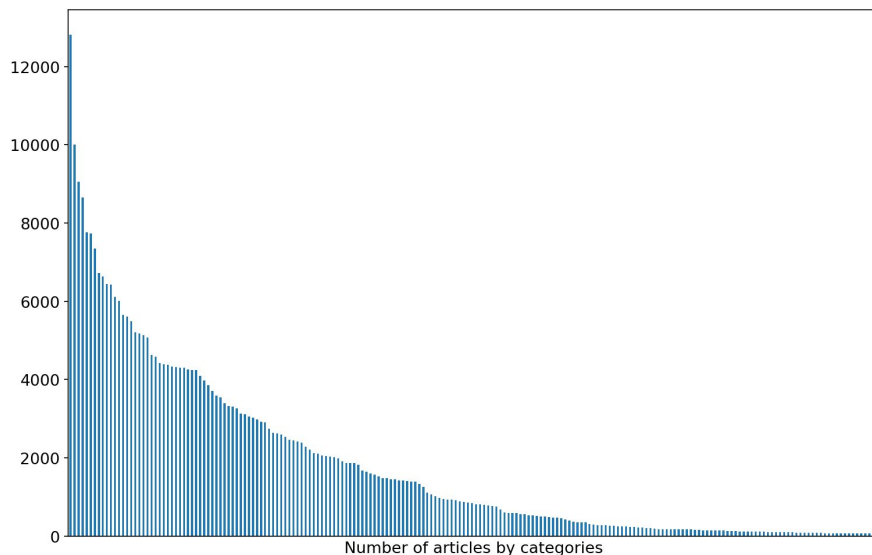


The screenshot shows the Kaggle dataset page for "News Portal User Interactions by Globo.com". The page features a dark header with the Kaggle logo, a search bar, and navigation links for Competitions, Datasets, Kernels, Discussion, and Learn. A "Sign In" button is in the top right. Below the header, the dataset title is prominently displayed, along with a subtitle: "A large dataset for news recommendations offline evaluation and analytics". The creator's name, Gabriel Moreira, and the update date are also visible. A "9 voters" badge is present on the right. The main content area has tabs for Data, Overview, Kernels, Discussion, and Activity. A "Download (351 MB)" button and a "New Kernel" button are located at the bottom right of the header section. Below this, there is a section for "Data (351 MB)" with an API link and a "Download All" button. The page is divided into three columns: "Data Sources" listing files like articles\_metadata.csv, clicks\_sample.csv, and clicks.zip; "About this file" providing a description of the metadata; and "Columns" listing fields such as article\_id, category\_id, created\_at\_ts, publisher\_id, and words\_count.

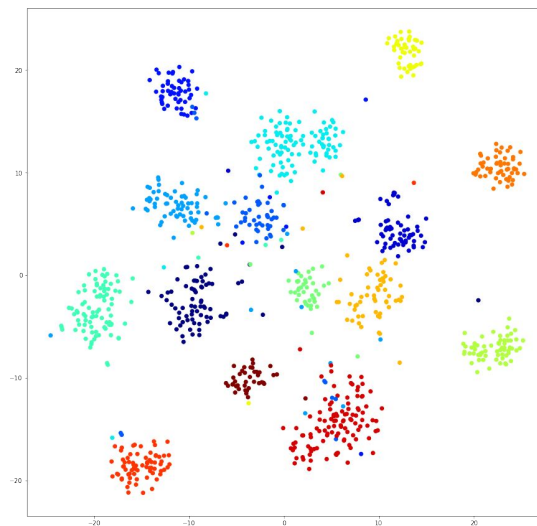
<https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom>

# ACR module training

Trained in a dataset with **364 K** articles from **461** categories, to generate the *Articles Content Embeddings* (vectors with 250 dimensions)

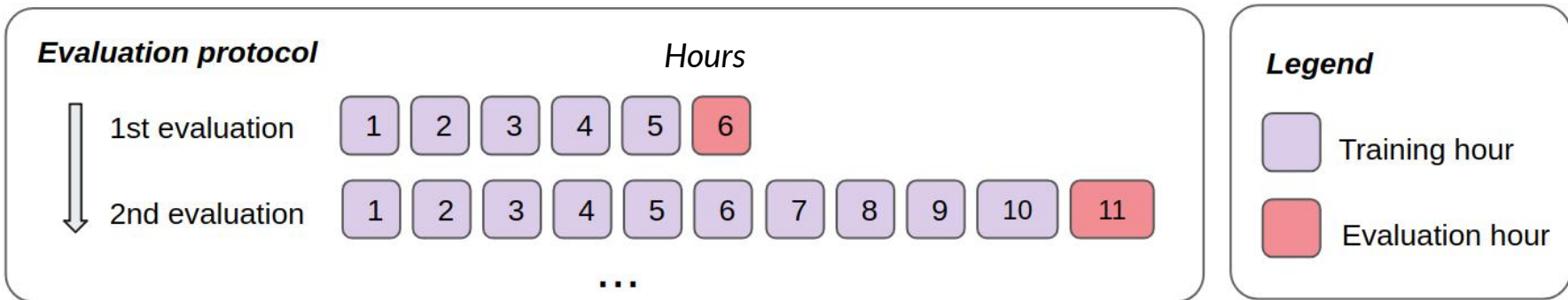


Distribution of articles by the top 200 categories



t-SNE visualization of trained *Article Content Embeddings* (from top 15 categories)

# Recommendation evaluation



**Task:** For each item within a session, predict the next-clicked item from a set composed by the positive sample (correct article) and 50 negative samples.

## Accuracy Metrics:

- **HitRate@10** - Checks whether the positive item is among the top-10 ranked items
- **MRR@10** - Ranking metric which assigns higher scores at top ranks.

# Recommendation evaluation

## Benchmark methods for session-based recommendations:

### Frequent patterns methods

1. **Co-occurrent** - Recommends articles commonly viewed together with the last read article, in other user sessions (simplified version of the association rules technique, with the maximum rule size of two) (Jugovac, 2018) (Ludewig, 2018)
2. **Sequential Rules (SR)** - A more sophisticated version of association rules, which considers the sequence of clicked items within the session. A rule is created when an item  $q$  appeared after an item  $p$  in a session, even when other items were viewed between  $p$  and  $q$ . The rules are weighted by the distance  $x$  (number of steps) between  $p$  and  $q$  in the session with a linear weighting function (Ludewig, 2018)

# Recommendation evaluation

## Baseline methods for session-based recommendations:

### KNN methods

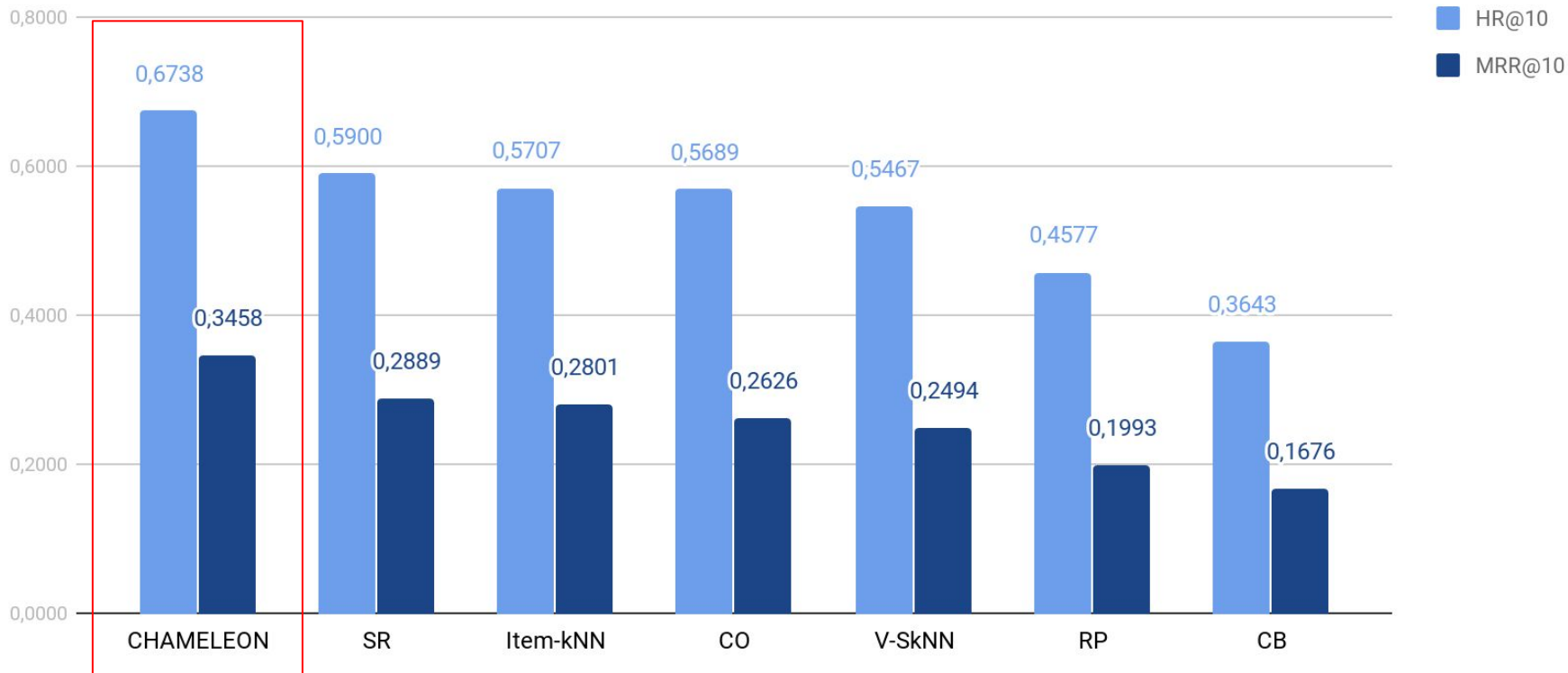
4. **Item-kNN** - Returns most similar items to the last read article, in terms of the cosine similarity between the vector of their sessions, i.e. it is the number of co-occurrences of two items in sessions divided by the square root of the product of the numbers of sessions in which the individual items are occurred.
5. **Vector Multiplication Session-Based kNN (V-SkNN)** - Compares the entire active session with past sessions and find items to be recommended. The comparison emphasizes items more recently clicked within the session, when computing the similarities with past sessions (Jannach,2017) (Jugovac,2018) (Ludewig,2018)

### Other baselines

6. **Recently Popular** - Recommends the most viewed articles from the last N clicks buffer
7. **Content-Based** - For each article read by the user, recommends similar articles based on the cosine similarity of their *Article Content Embeddings*, from the last N clicks buffer.

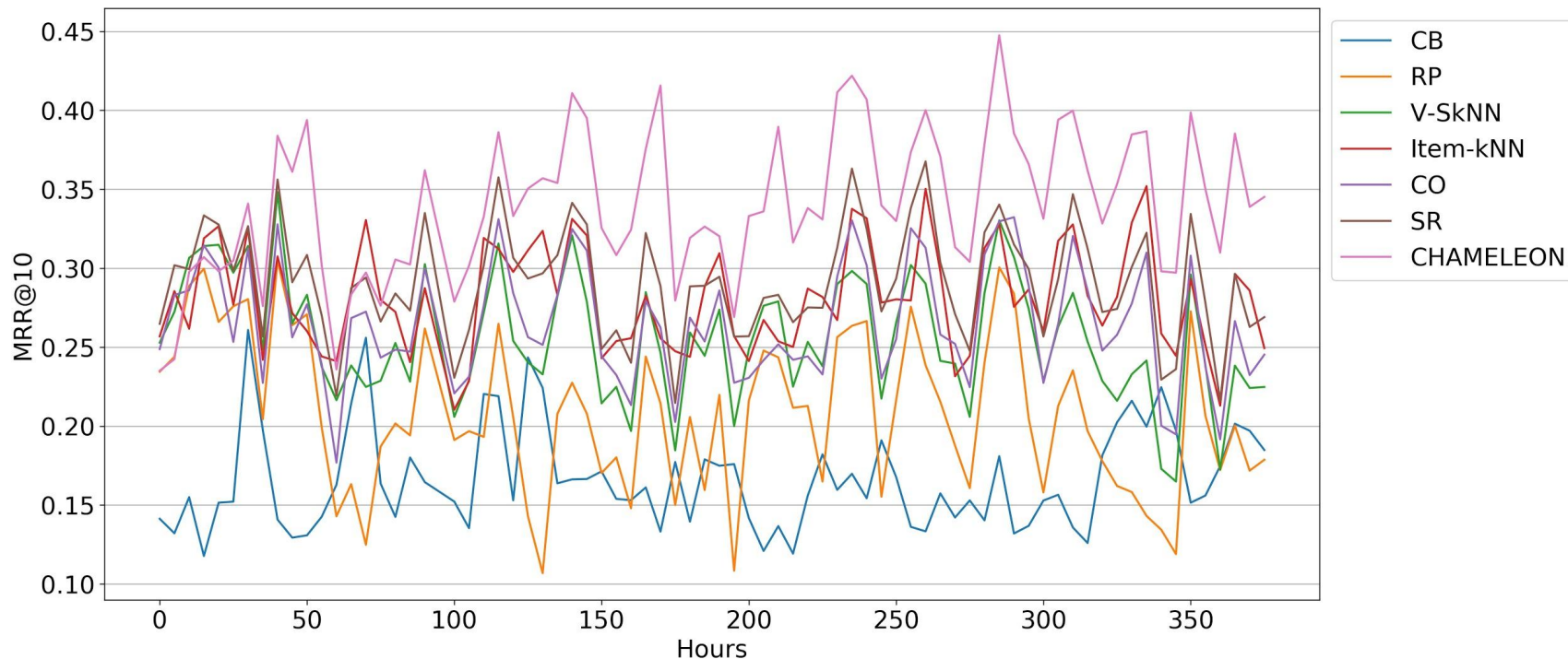
# Recommendation evaluation

## Recommendation Accuracy Metrics



# Recommendation evaluation

Continuous training and evaluating during 16 days (Oct. 1-16, 2017)



Average **MRR@10** by hour (evaluation each 5 hours)



# Recommendation evaluation

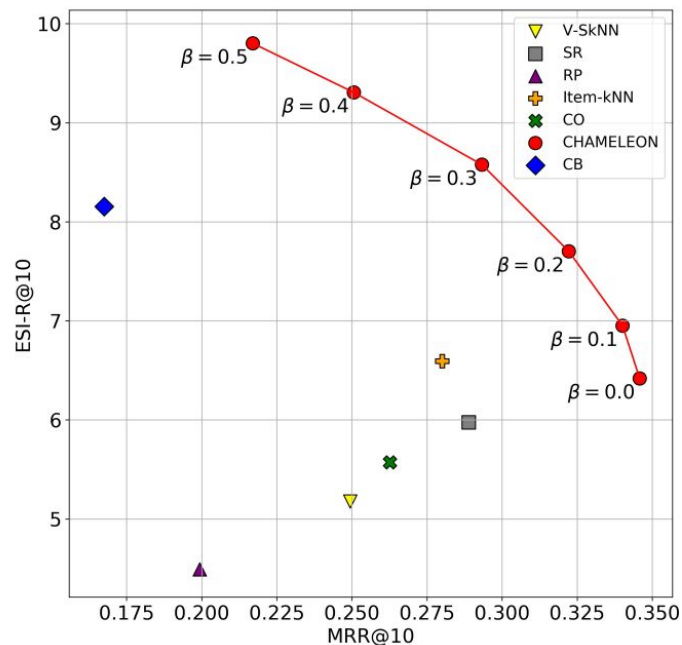
## *Other recommendation quality factors*

<i>Recommender</i>	<i>Item Coverage</i> <i>COV@10</i>	<i>Novelty</i> <i>ESI-R@10</i>	<i>Diversity</i> <i>EILD-R@10</i>
<b>G1 dataset</b>			
<i>CHAMELEON</i>	0.6373	6.4177	0.3620
<i>SR</i>	0.2763	5.9747	0.3526
<i>Item-kNN</i>	0.3913	6.5909	0.3552
<i>CO</i>	0.2499	5.5728	0.3570
<i>V-SkNN</i>	0.1355	5.1760	0.3558
<i>RP</i>	0.0218	4.4904	<b>0.3750</b>
<i>CB</i>	<b>0.6774</b>	<b>8.1531</b>	0.2789

# Recommendation evaluation

## *Balancing conflicting objectives*

$$L(\theta) = \text{accuracy\_loss}(\theta) - \beta * \text{nov\_loss}(\theta)$$



## News Session-Based Recommendations using Deep Neural Networks

Gabriel de Souza Pereira  
Moreira\*  
CI&T  
Campinas, SP, Brazil  
gabrielpm@ciandt.com

Felipe Ferreira  
Globo.com  
Rio de Janeiro, RJ, Brazil  
felipe.ferreira@corp.globo.com

Adilson Marques da Cunha  
Brazilian Aeronautics Institute of  
Technology - ITA  
São José dos Campos, SP, Brazil  
cunha@ita.br

### ABSTRACT

News recommender systems are aimed to personalize users experiences and help them to discover relevant articles from a large and dynamic search space. Therefore, news domain is a challenging scenario for recommendations, due to its sparse user profiling, fast growing number of items, accelerated item's value decay, and users preferences dynamic shift.

Some promising results have been recently achieved by the usage of Deep Learning techniques on Recommender Systems, specially for item's feature extraction and for session-based recommendations with Recurrent Neural Networks.

In this paper, it is proposed an instantiation of the CHAMELEON – a Deep Learning Meta-Architecture for News Recommender Systems. This architecture is composed of two modules, the first responsible for item's feature extraction, and the second, for session-based recommendations.

### CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies → Neural networks;

### KEYWORDS

Recommender Systems; Deep Learning; News Recommendation; Session-Based Recommendation; Context-Based Recommendation; Recurrent Neural Networks

### ACM Reference Format:

Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News Session-Based Recommendations Networks. In *3rd Workshop on Deep Learning for Recommendation (DLRS 2018)*, October 6, 2018, Vancouver, BC, Canada. ACM, USA, 9 pages. <https://doi.org/10.1145/3270323.3270328>

<https://arxiv.org/abs/1904.10367>

v1 [cs.IR] 15 Apr 2019

## Contextual Hybrid Session-based News Recommendation with Recurrent Neural Networks

Gabriel de Souza P. Moreira<sup>a,b</sup>, Dietmar Jannach<sup>c</sup>, Adilson Marques da Cunha<sup>a</sup>

<sup>a</sup>Department of Electrical Engineering and Computing, Instituto Tecnológico de Aeronáutica (ITA) - São José dos Campos, São Paulo, Brazil

<sup>b</sup>CI&T - Campinas, São Paulo, Brazil

<sup>c</sup>Department of Applied Informatics, University of Klagenfurt, Austria

### Abstract

Recommender systems help users deal with information overload by providing tailored item suggestions to them. The recommendation of news is often considered to be challenging, since the relevance of an article for a user can depend on a variety of factors, including the user's short-term reading interests, the reader's context, or the recency or popularity of an article.

Previous work has shown that the use of Recurrent Neural Networks is

<https://arxiv.org/abs/1808.00076>

An underwater photograph showing two divers on jet skis. The diver in the foreground is on a yellow and black jet ski, while the second diver is on a red and black one. They are swimming over a rocky seabed with some coral. The water is clear and blue.

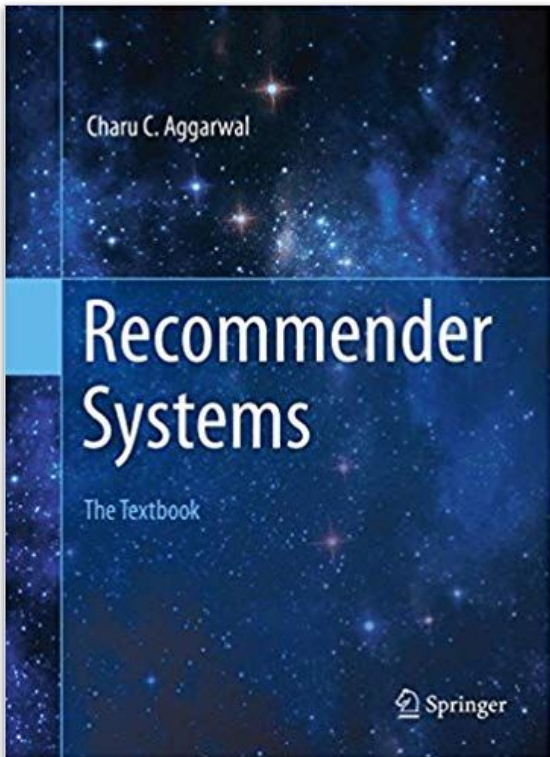
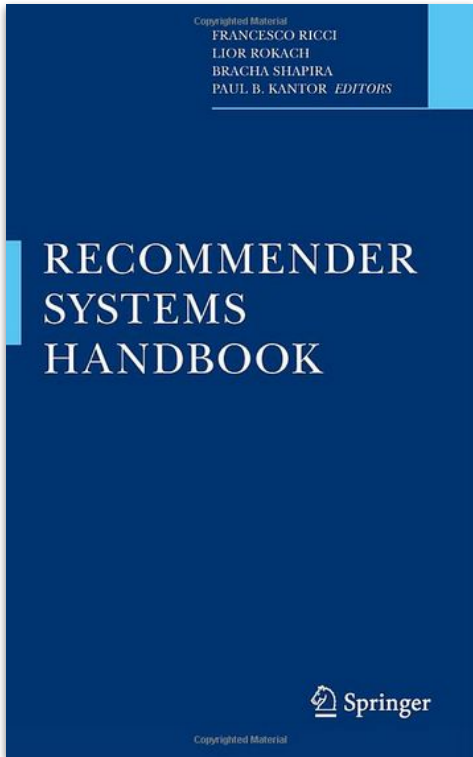
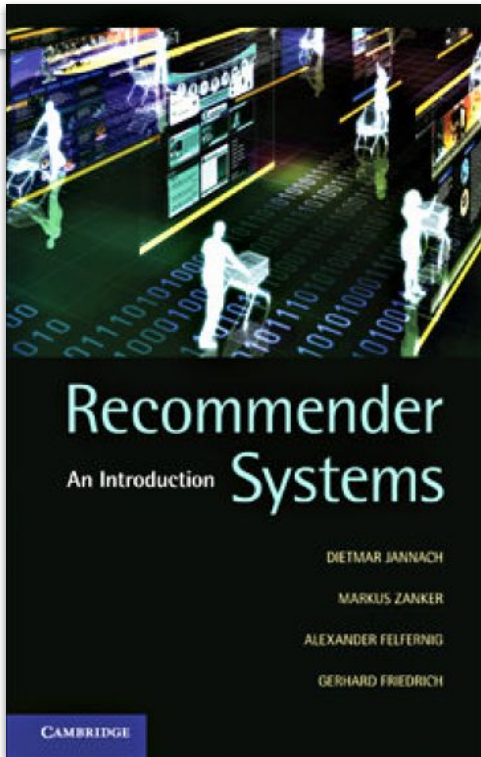
# Deep Recommender Systems

 **PAPIS.io**  
LatAm 2018

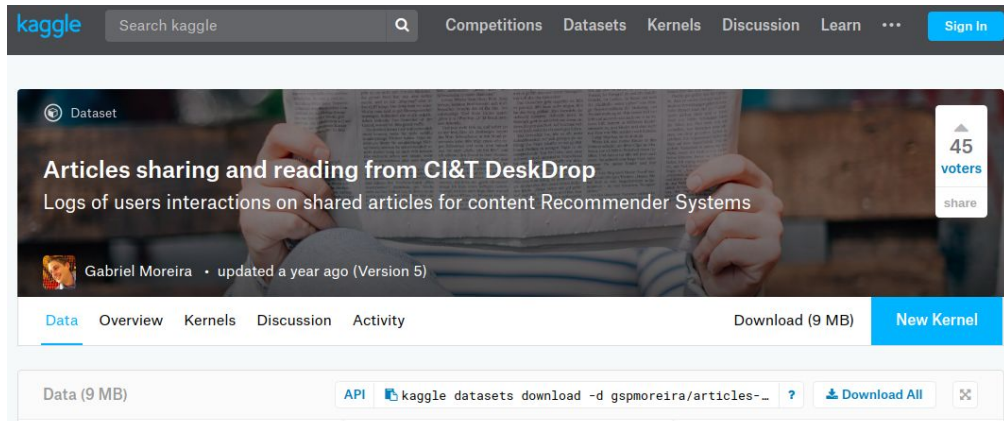
**Gabriel Moreira** • @gspmoreira  
Lead Data Scientist      Phd. Candidate

# References



# CI&T Deskdrop dataset on Kaggle!



The screenshot shows the Kaggle website interface. At the top, there is a search bar with 'kaggle' and 'Search kaggle' text, and navigation links for 'Competitions', 'Datasets', 'Kernels', 'Discussion', 'Learn', and a 'Sign In' button. The main content area features a dataset card for 'Articles sharing and reading from CI&T DeskDrop' by Gabriel Moreira, updated a year ago (Version 5). The card includes a description: 'Logs of users interactions on shared articles for content Recommender Systems', a '45 voters' badge, and a 'share' button. Below the card, there are tabs for 'Data', 'Overview', 'Kernels', 'Discussion', and 'Activity', along with 'Download (9 MB)' and 'New Kernel' buttons. At the bottom, there is a terminal-like interface showing the command 'kaggle datasets download -d gspmoreira/articles-...' and a 'Download All' button.

- 12 months logs (Mar. 2016 - Feb. 2017)
- ~ 73k logged users interactions
- ~ 3k public articles shared in the platform.

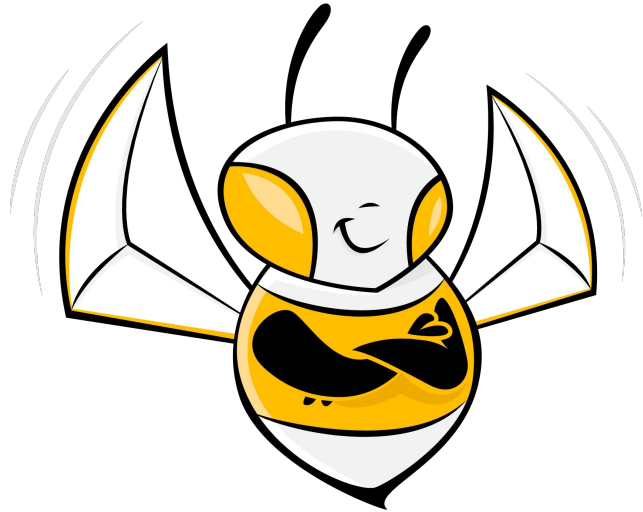
<https://www.kaggle.com/gspmoreira/articles-sharing-reading-from-cit-deskdrop>

## Recommender Systems in Python 101

<https://www.kaggle.com/gspmoreira/recommender-systems-in-python-101>



TDC 2019



**MACHINE**  
**LEARNING**  
CI&T

# Questions?

**Gabriel Moreira**

*Lead Data Scientist*  
*[gabrielpm@ciandt.com](mailto:gabrielpm@ciandt.com)*  
*[@gspmoreira](https://twitter.com/gspmoreira)*

