TDC 2019

Gabriel Moreira
Lead Data Scientist
CI&T

Deep Learning for Recommender Systems

About me

Gabriel Moreira

@gspmoreira

CI&T

Lead Data Scientist

ITA

DSc. candidate

Google Developers Experts 2019

Machine Learning GDE

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

Partner

Google Cloud

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

38% of sales

NETFLIX

75% of watched content

Google news

39% of top news visualization

What else may I recommend?

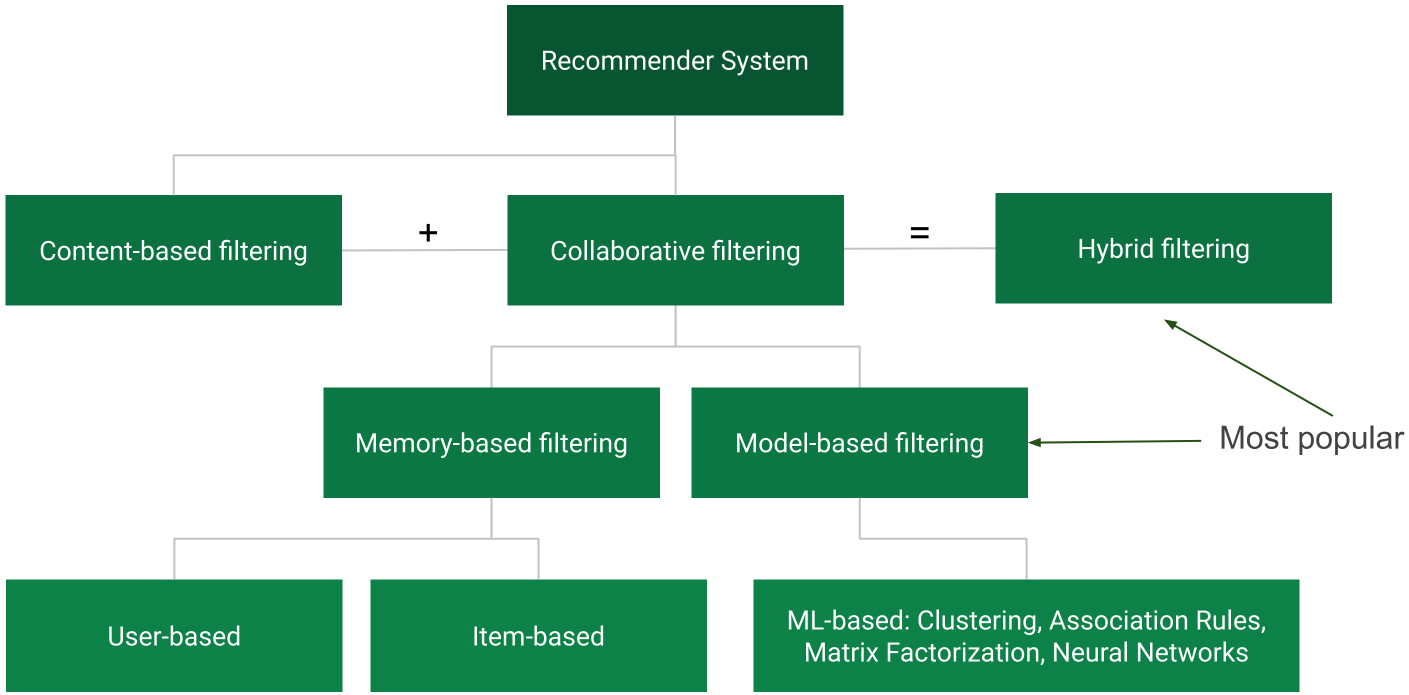
products
tags
professionals
courses
musics movies
jobs books
papers girlfriends
investiments restaurants
videos
dressing

What can a Recommender Systems do?

1 - Recommendation
Given a user, produce an ordered list matching the user needs

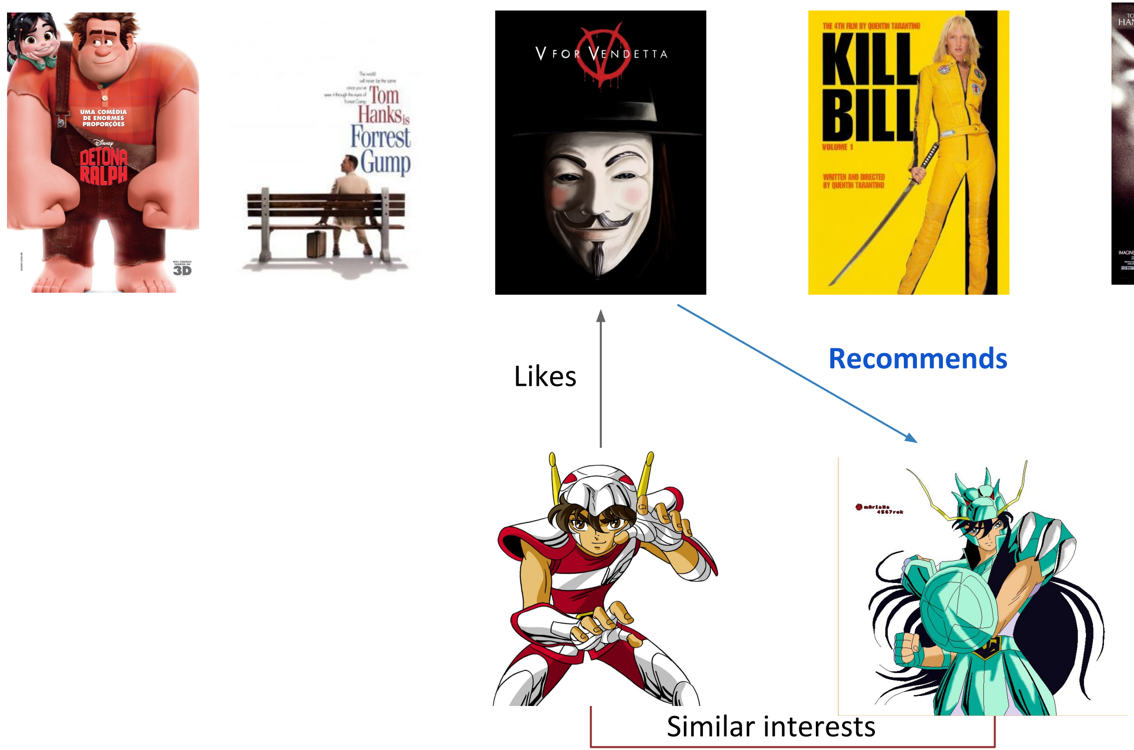
2 - Prediction
Given an item, what is its relevance for each user?

Recommender System Methods

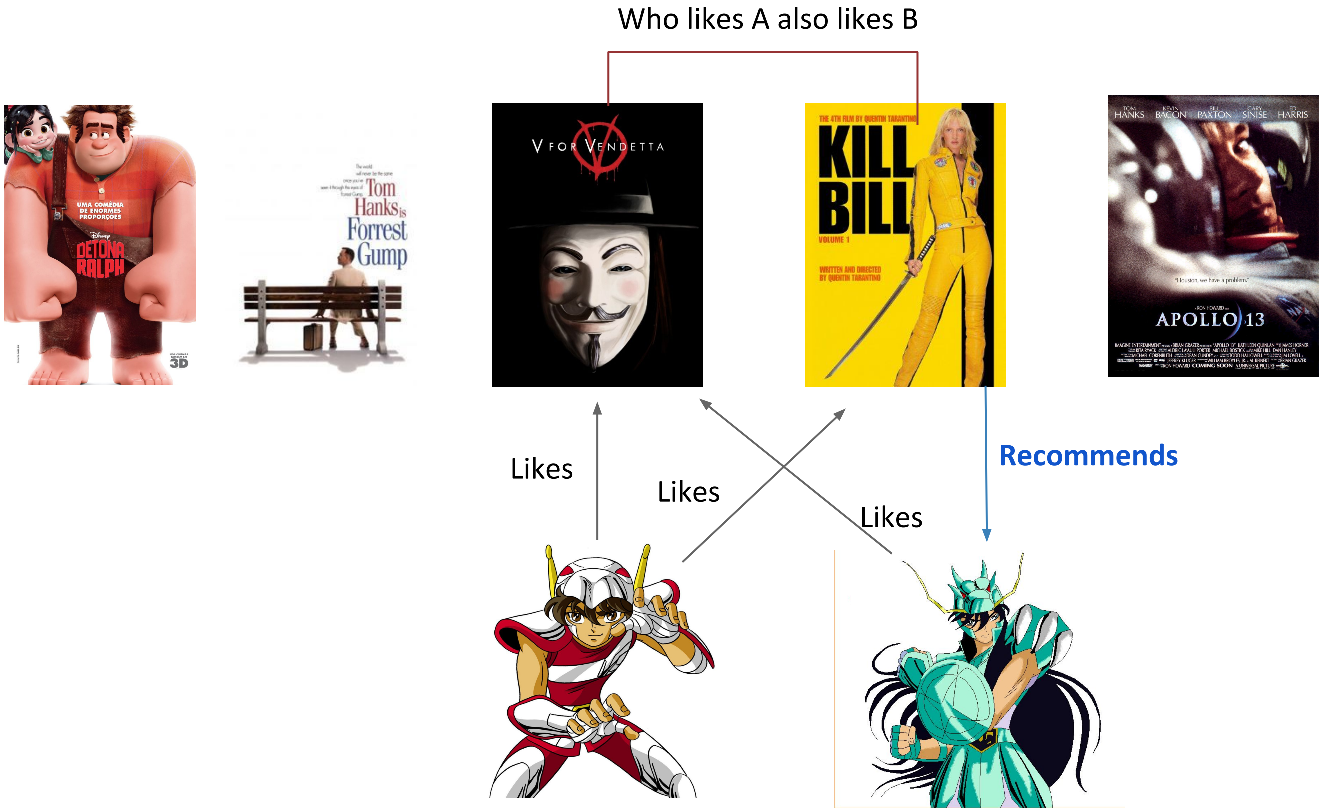


Collaborative Filtering

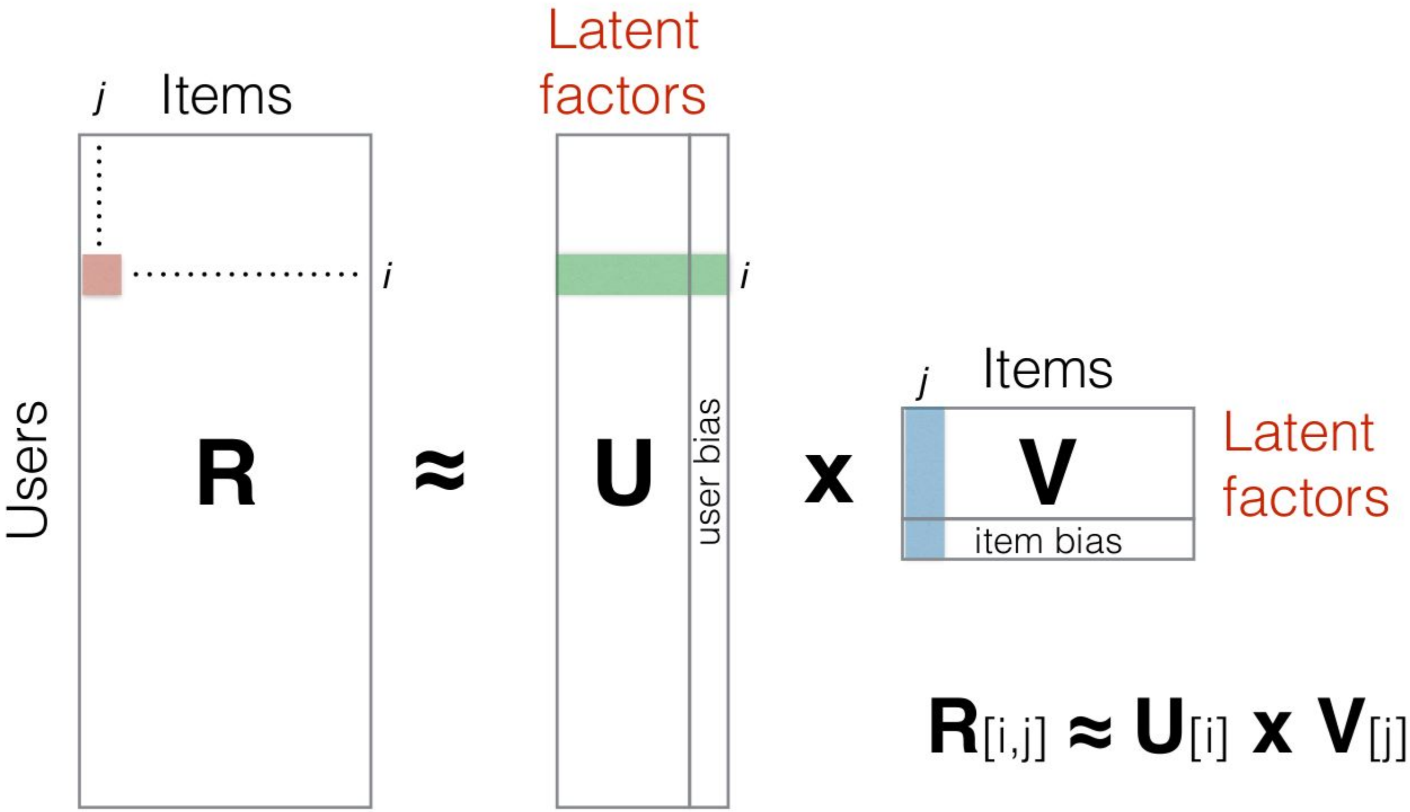
User-Based Collaborative Filtering



Item-Based Collaborative Filtering



Collaborative Filtering based on Matrix Factorization



$\mathbf{R}_{[i,j]} \approx \mathbf{U}_{[i]} \ \mathbf{x} \ \mathbf{V}_{[j]}$

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

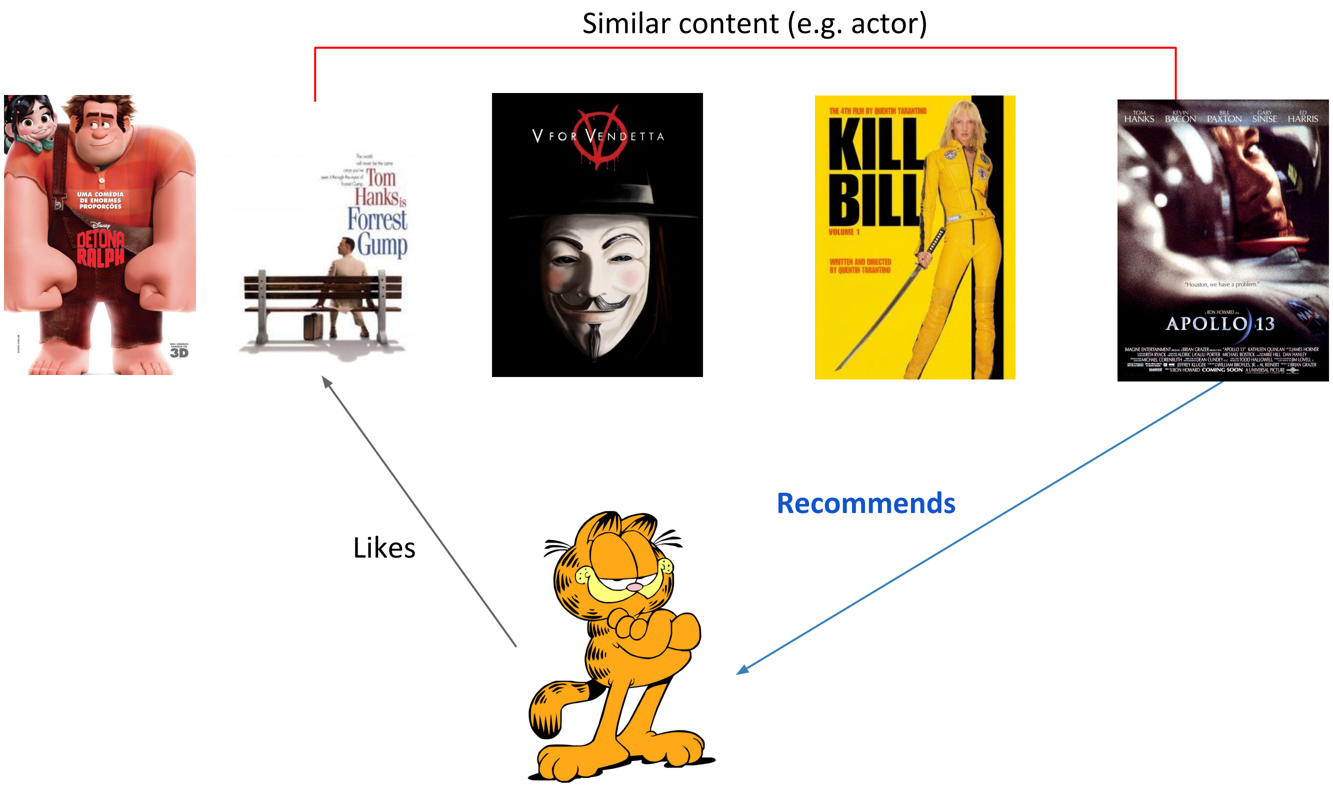
Python

Java

.NET

Content-Based Filtering

Content-Based Filtering



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.

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

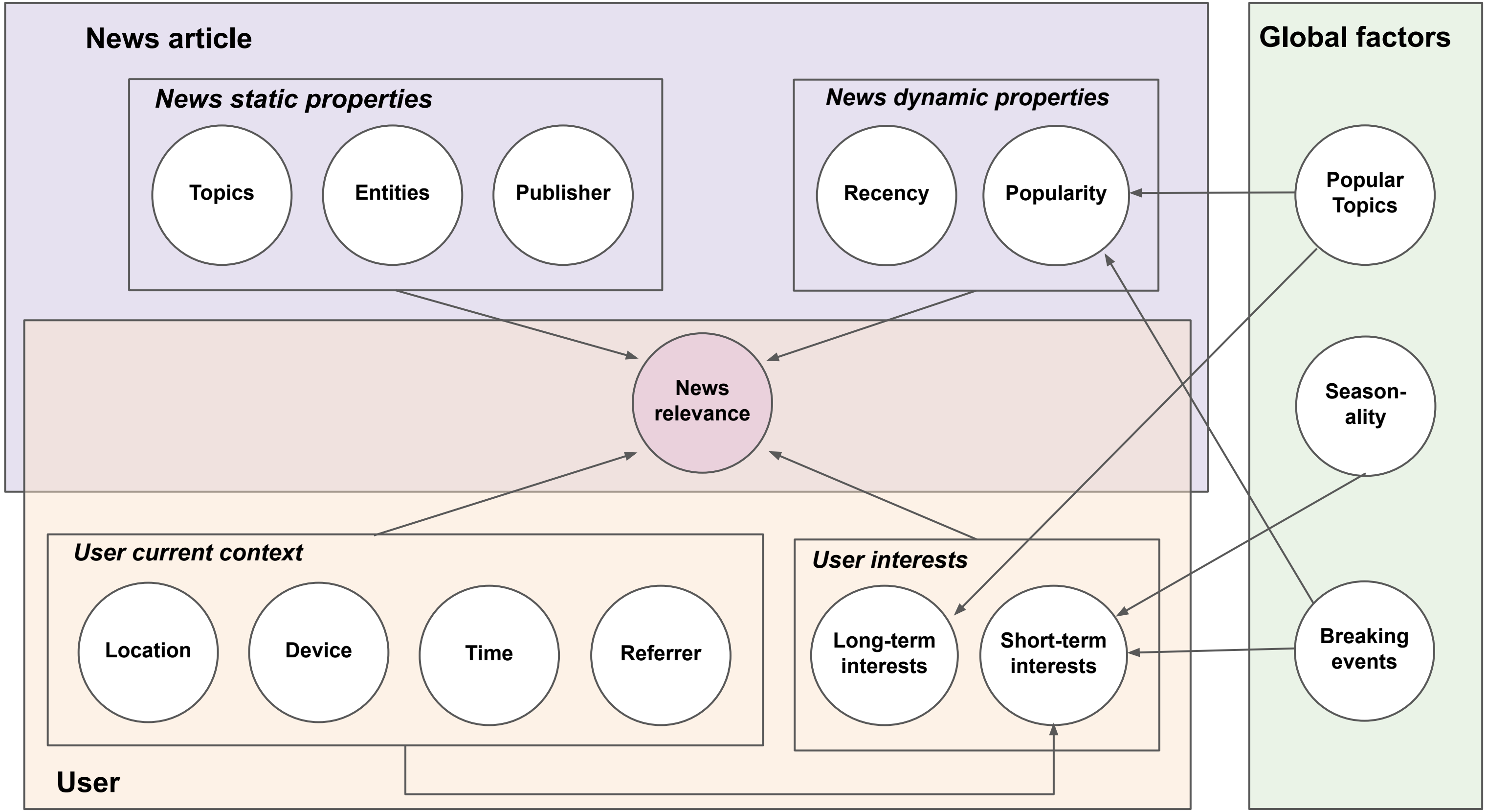
The majority of web traffic (TREVISIOL et al. , 2014b)

Challenges

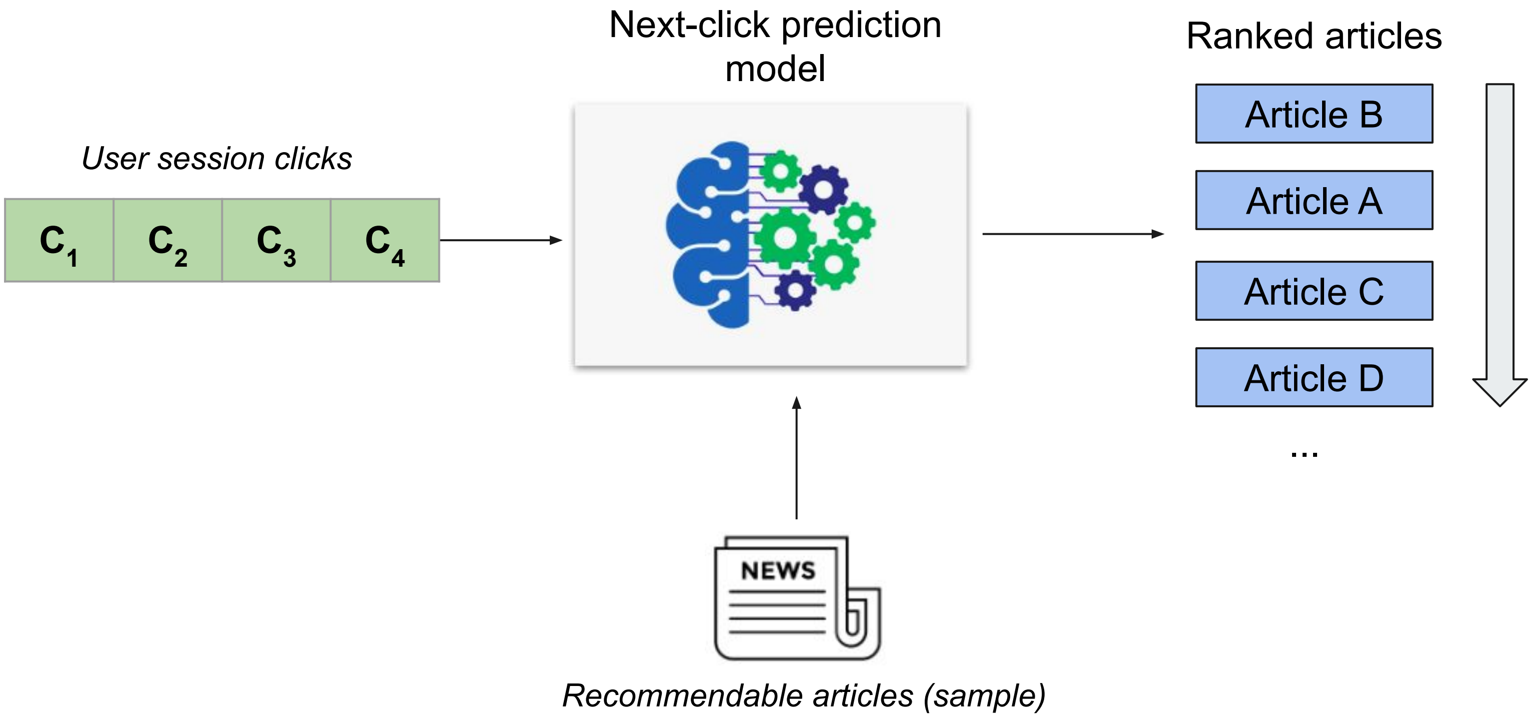
1. Streaming clicks and news articles
2. Most users are anonymous
3. Users' preferences shift
4. Accelerated relevance decay

Percentile of clicks	*Article age*
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

Factors affecting news relevance

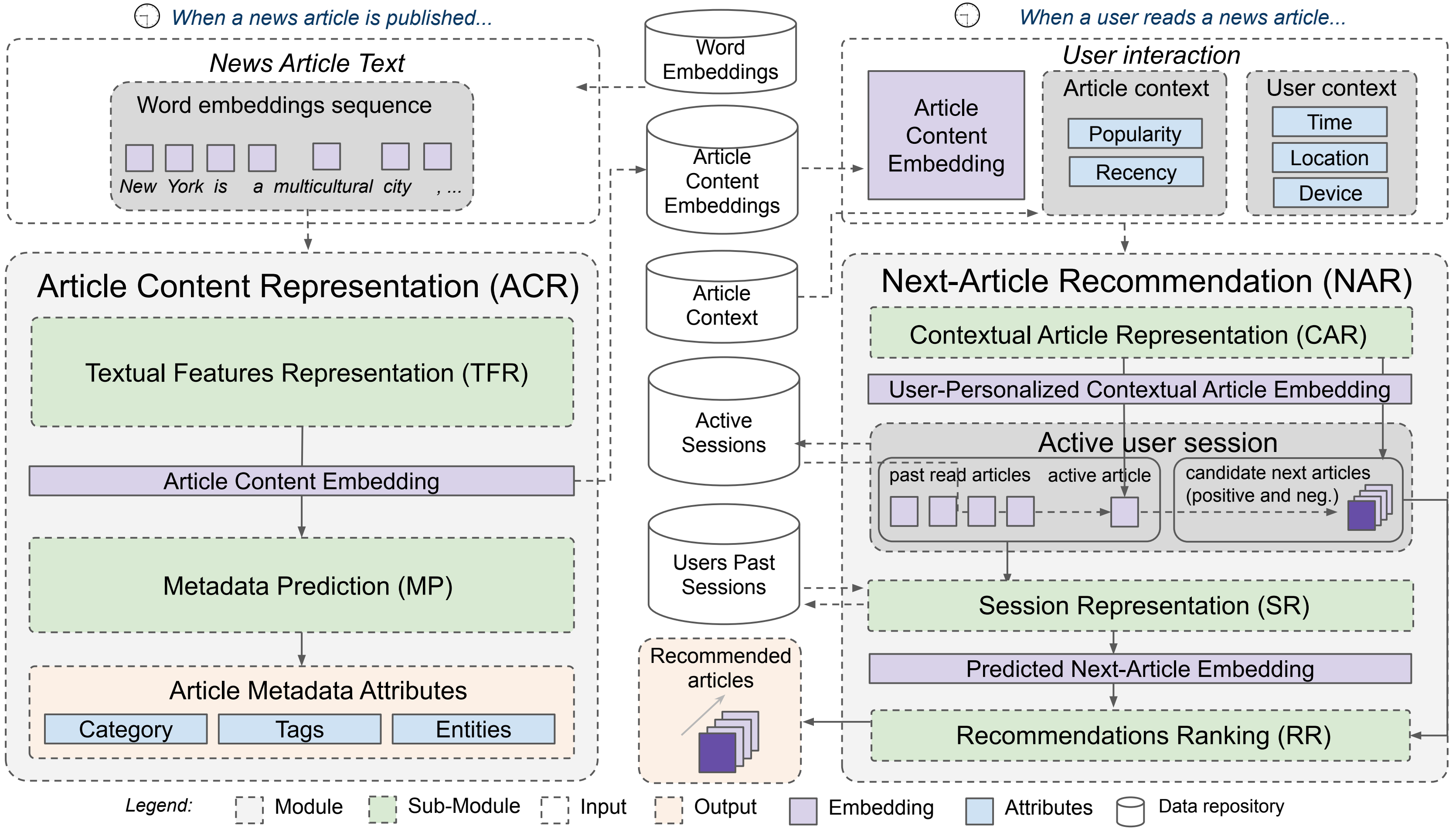


News session-based recommender overview

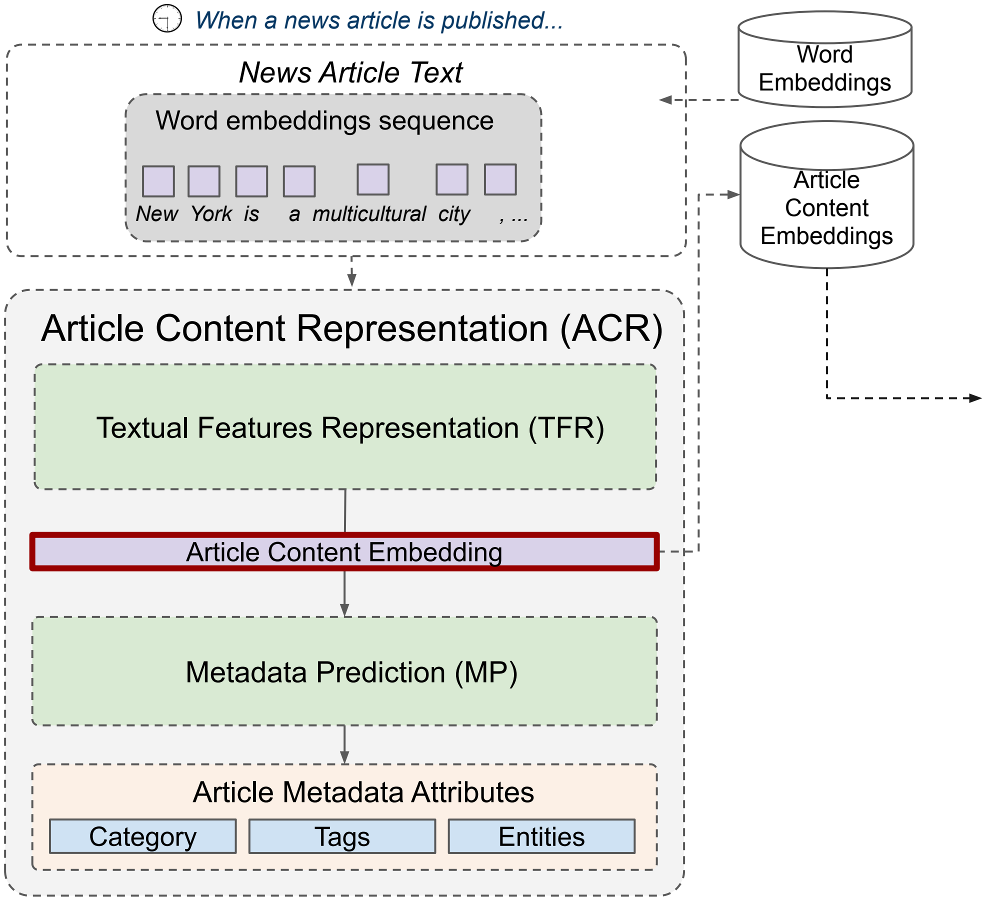


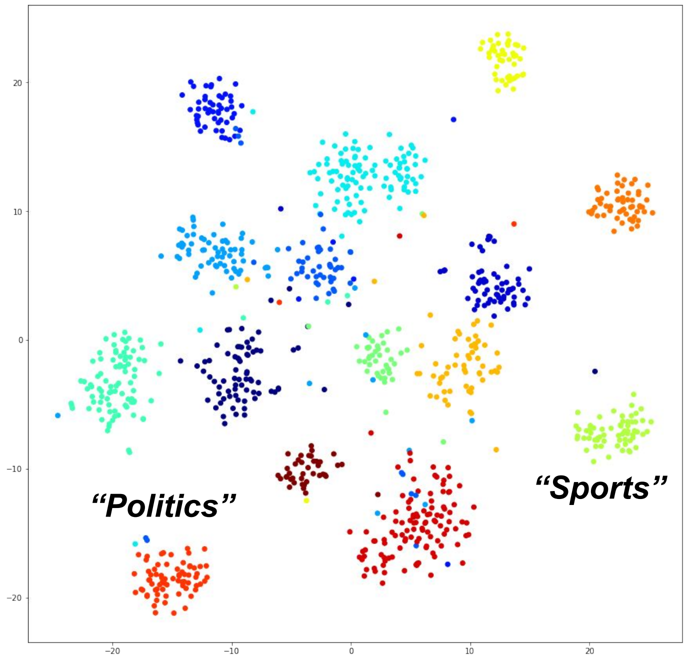
CHAMELEON: A Deep Learning Meta-Architecture for News Recommendation

CHAMELEON Meta-Architecture for News RS



CHAMELEON - ACR module

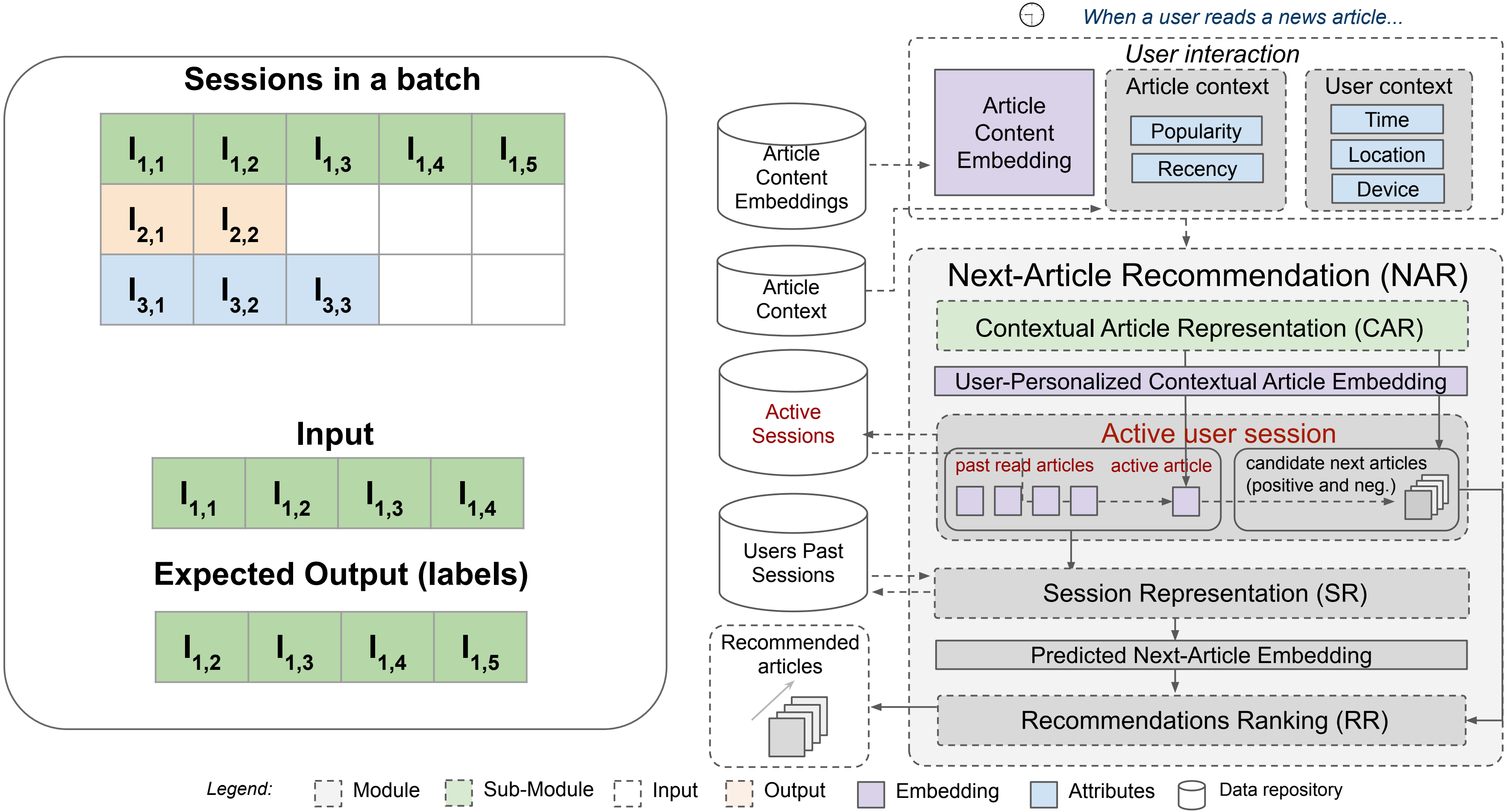




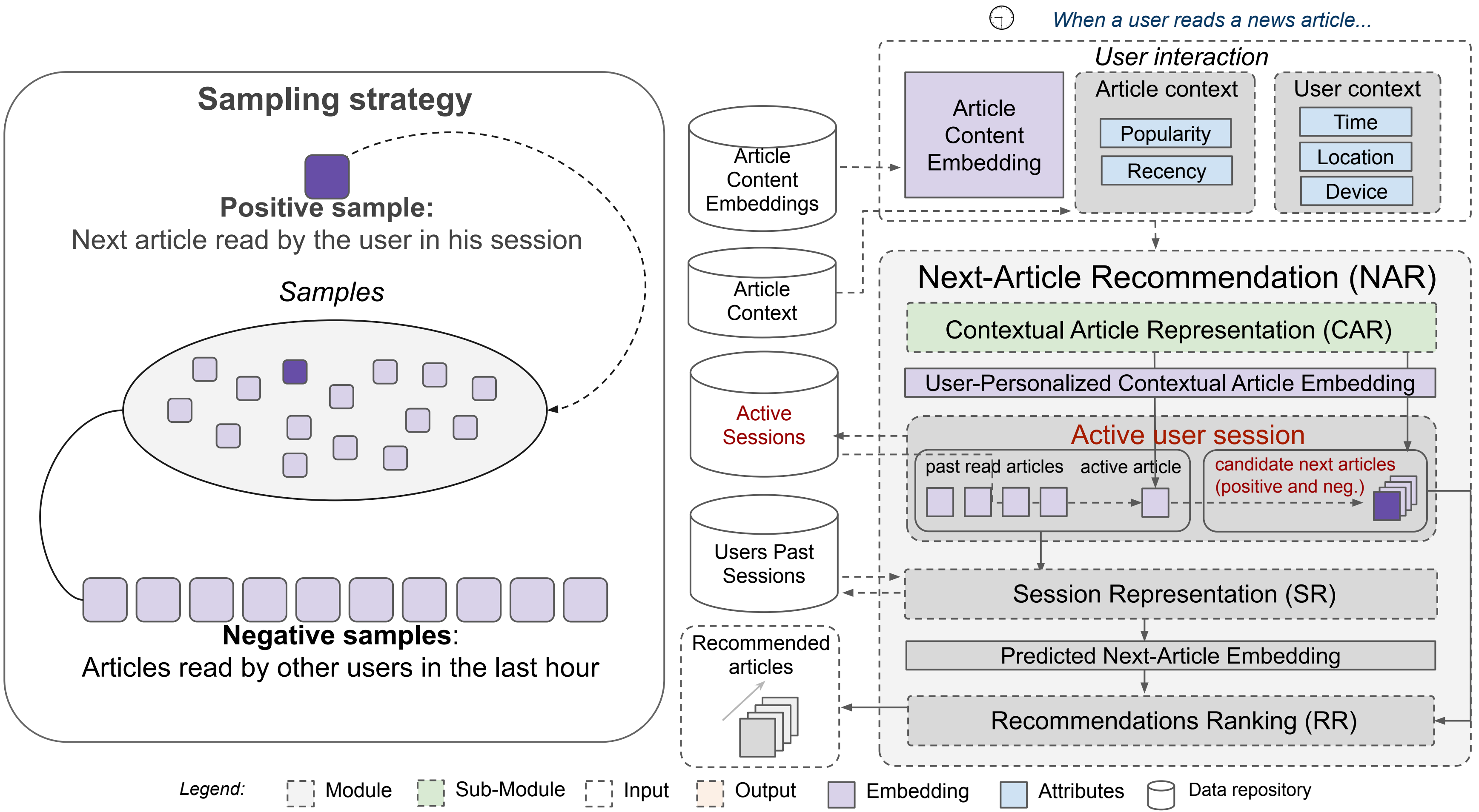
T-SNE viz. of articles embeddings colored by category, with similar articles highlighted

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

CHAMELEON - NAR module



CHAMELEON - NAR module

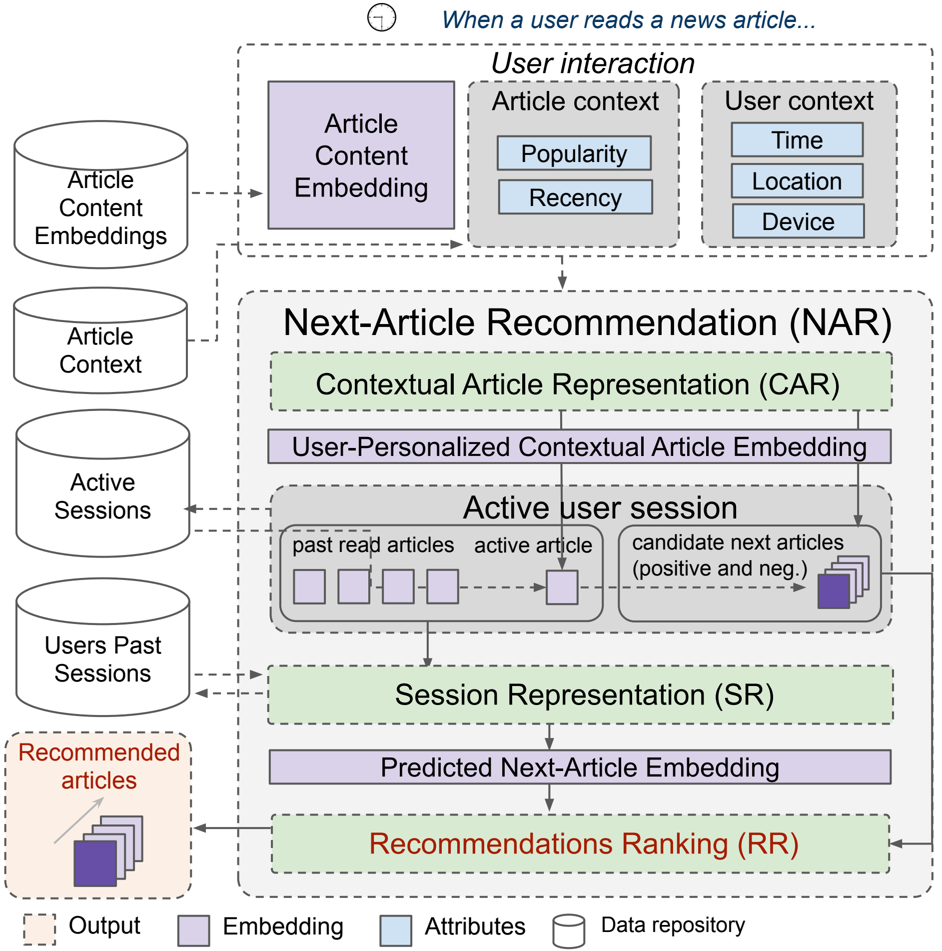


CHAMELEON - NAR module

Recommendations Ranking (RR) sub-module

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

Relevance Score of an item for a user session



CHAMELEON - Ranking loss function

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

Cosine similarity

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

Softmax over Relevance Score

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

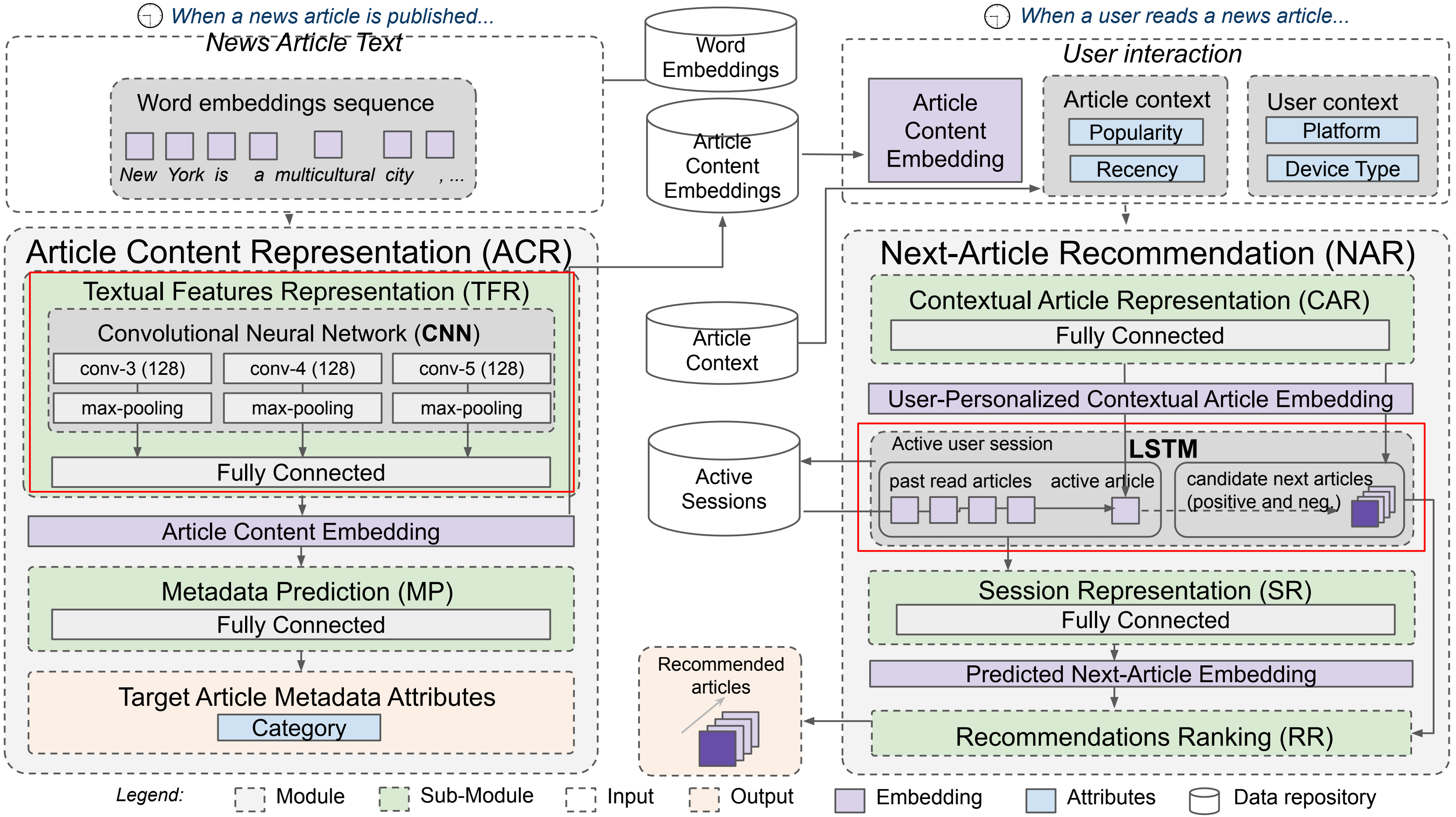
Loss function

```
#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

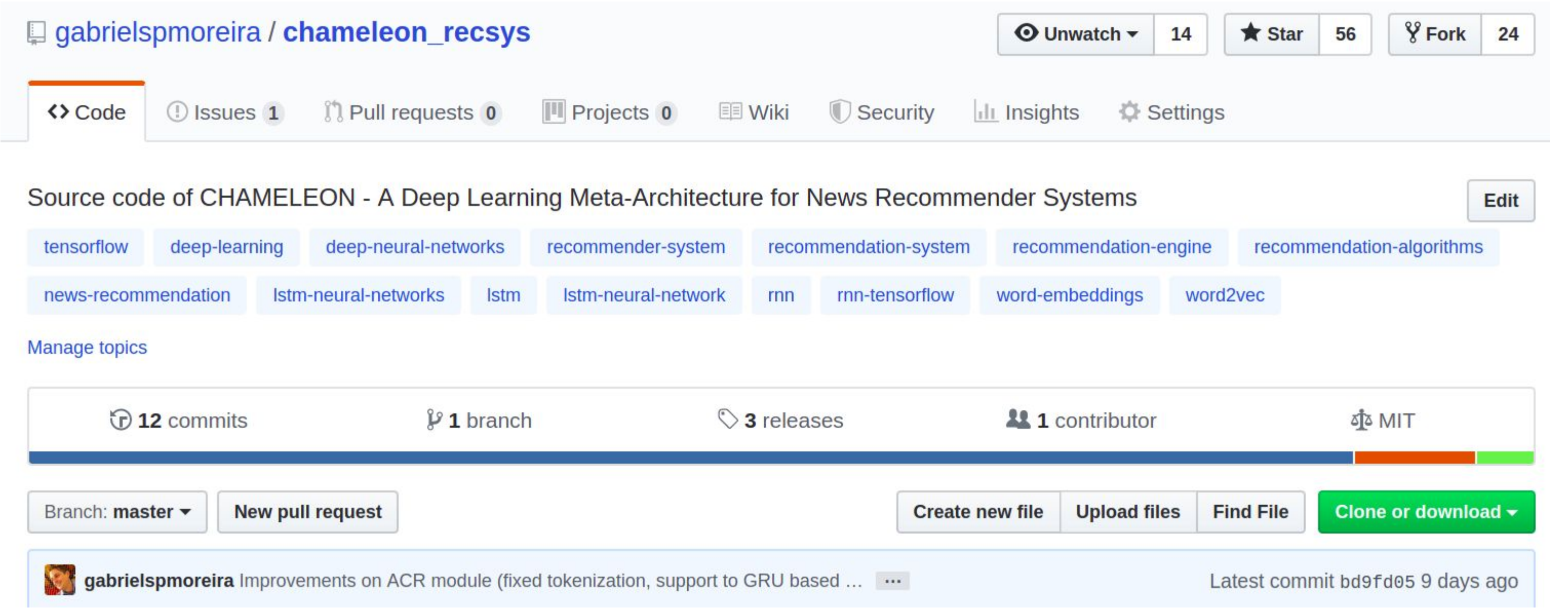
CHAMELEON
Architecture Instatiations

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

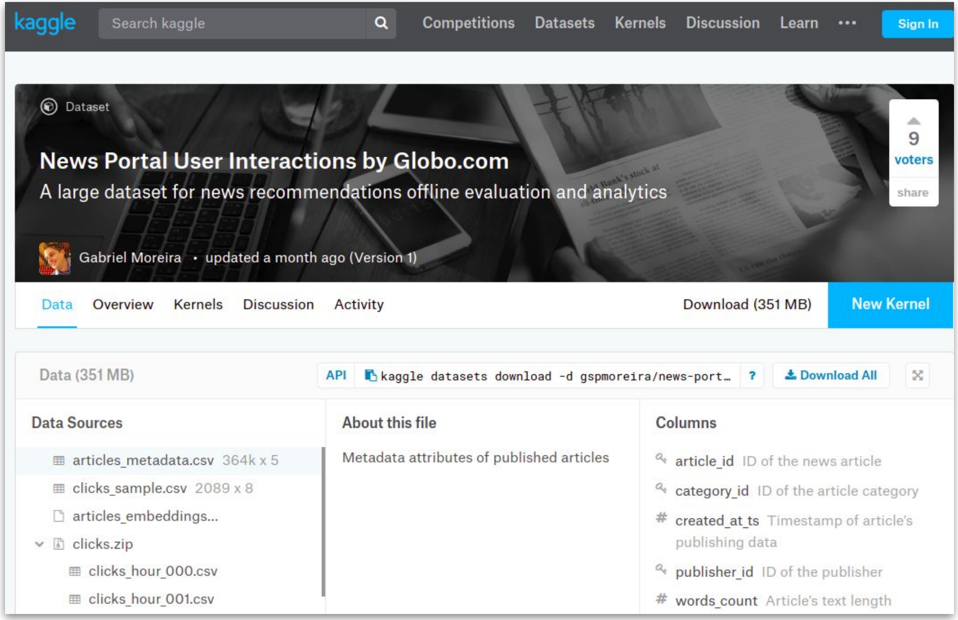


- Training and evaluation performed in **Google Cloud Platform ML Engine**

Experiments

Experiments - Dataset

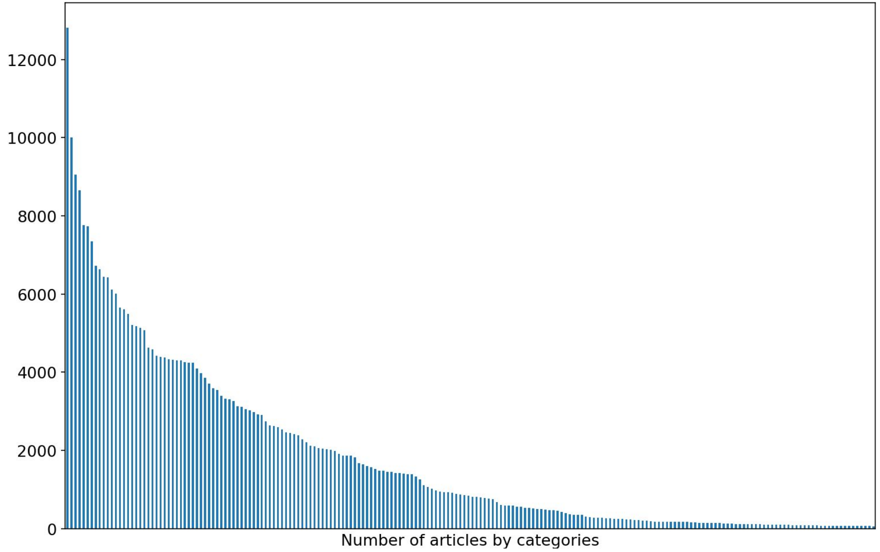
- Provided by **Globo.com (G1)**, the most popular news portal in Brazil
- Sample from **Oct., 1 to 16, 2017**, with over **3 M** clicks, distributed in **1.2 M sessions** from **330 K** users, who read over **50 K** unique news articles



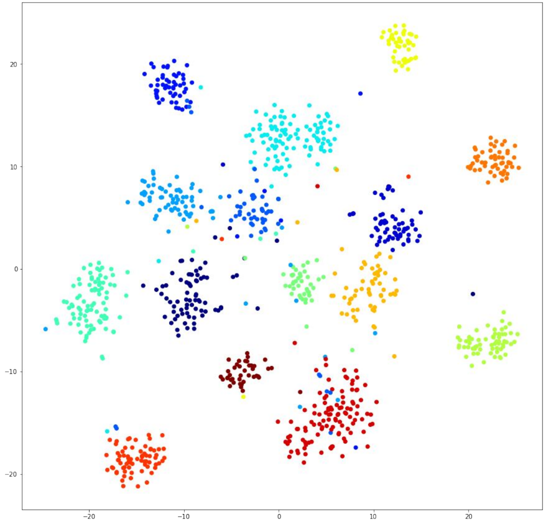
https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom

ACR module training

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

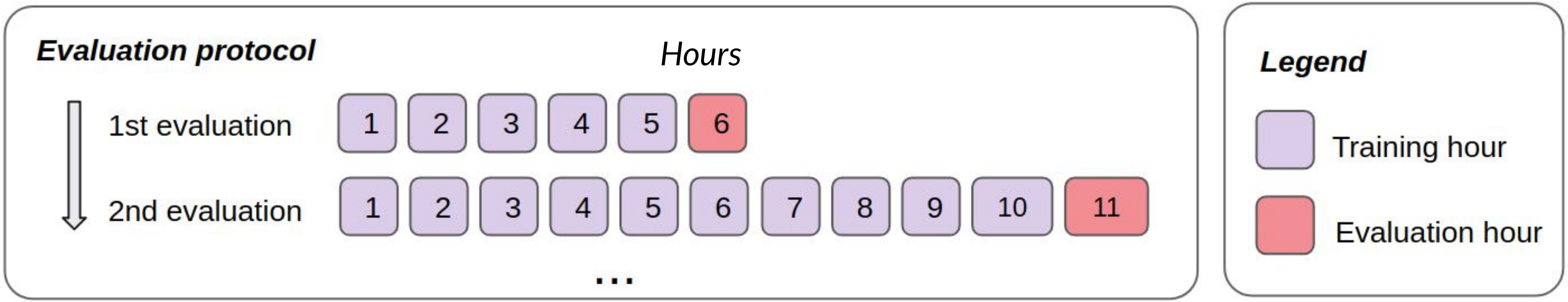


Distribution of articles by the top 200 categories



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:

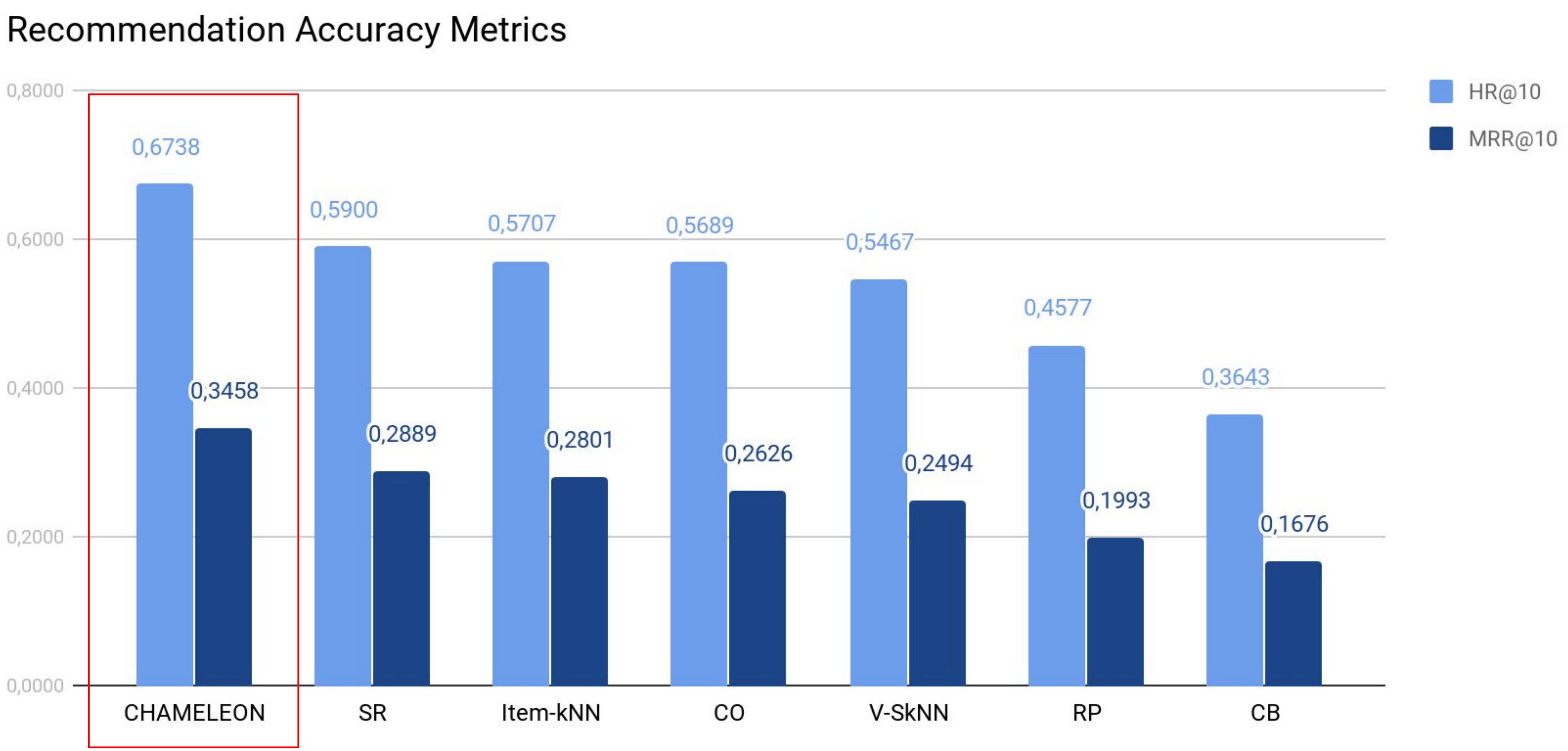
KNN methods

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

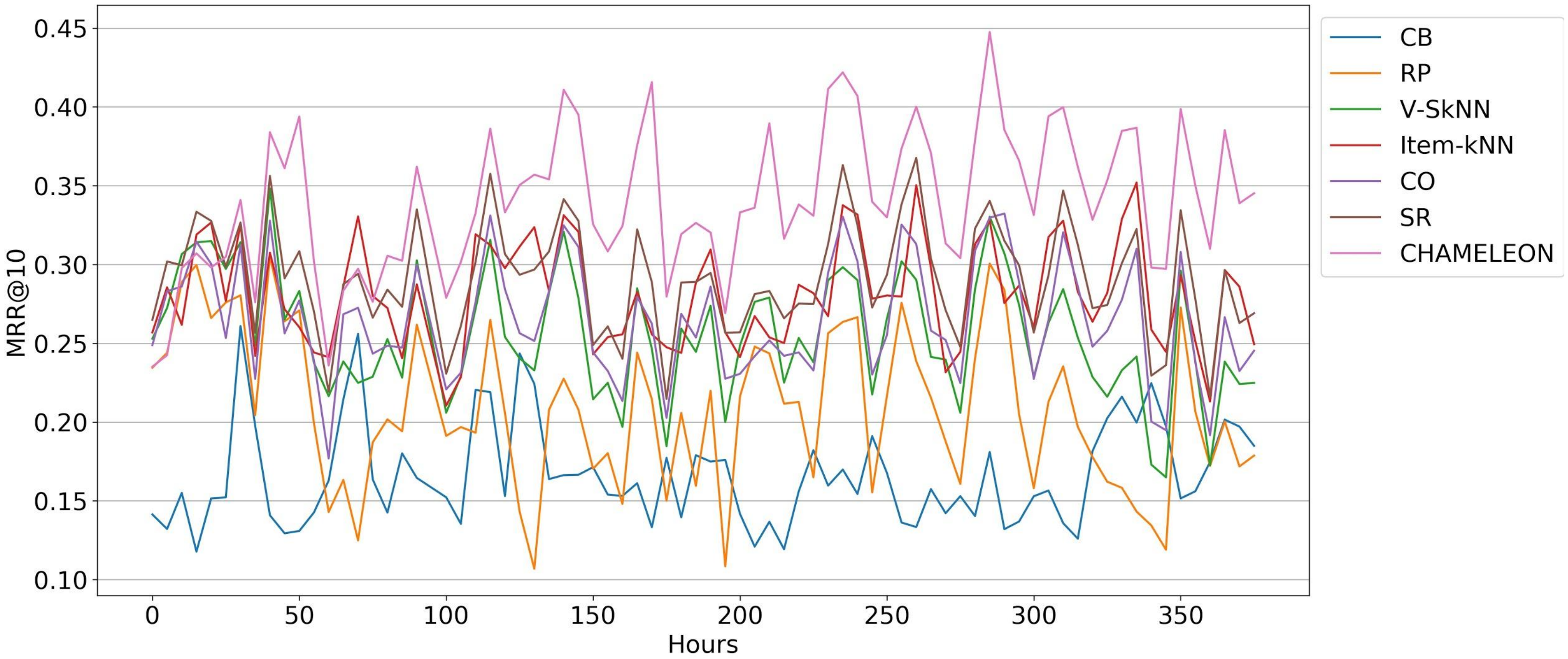
6. **Recently Popular** - Recommends the most viewed articles from the last N clicks buffer
7. **Content-Based** - For each article read by the user, recommends similar articles based on the cosine similarity of their *Article Content Embeddings*, from the last N clicks buffer.

Recommendation evaluation



Recommendation evaluation

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



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

Recommendation evaluation

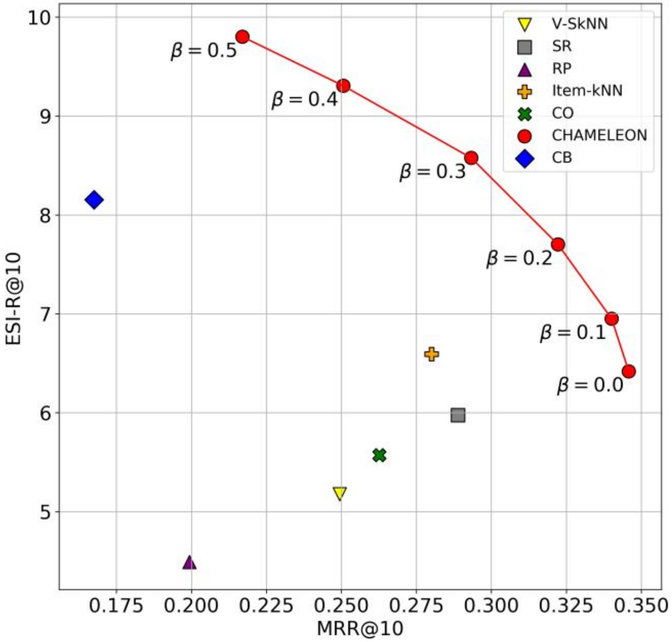
Other recommendation quality factors

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

Recommendation evaluation

Balancing conflicting objectives

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



References

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 respon-

CCS CONCEPTS

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

KEYWORDS

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

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

https://arxiv.org/abs/1904.10367

Contextual Hybrid Session-based News Recommendation with Recurrent Neural Networks

Gabriel de Souza P. Moreira[a,b], Dietmar Jannach[c], Adilson Marques da Cunha[a]

[a]*Department of Electrical Engineering and Computing, Instituto Tecnológico de Aeronáutica (ITA) - São José dos Campos, São Paulo, Brazil*
[b]*CI&T - Campinas, São Paulo, Brazil*
[c]*Department of Applied Informatics, University of Klagenfurt, Austria*

Abstract

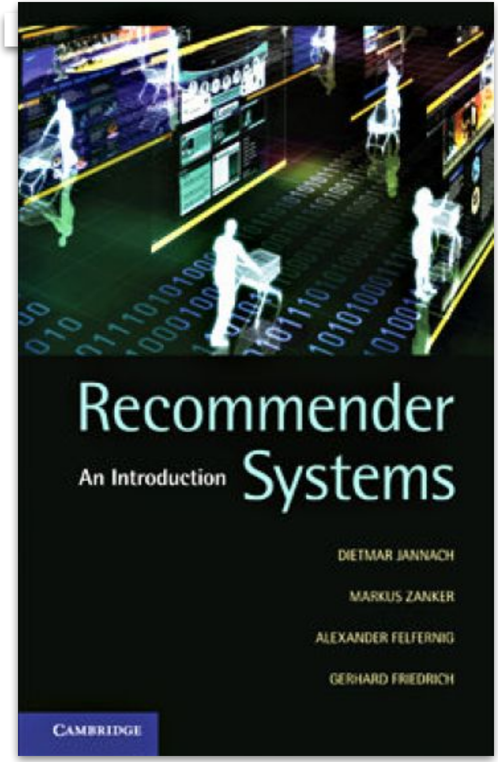
Recommender systems help users deal with information overload by providing tailored item suggestions to them. The recommendation of news is often considered to be challenging, since the relevance of an article for a user can depend on a variety of factors, including the user's short-term reading interests,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

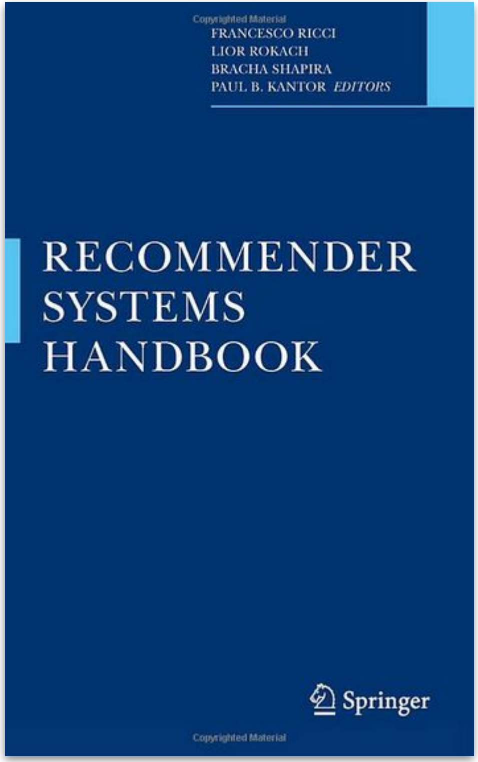
https://arxiv.org/abs/1808.00076

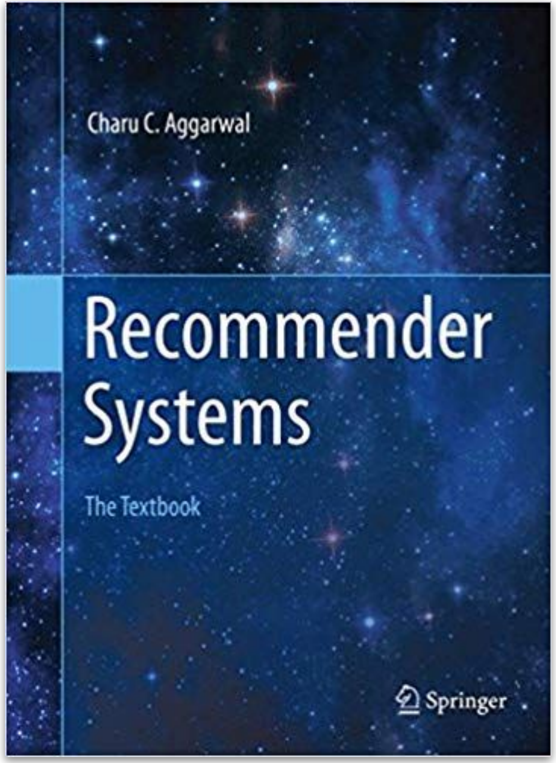
References

https://www.infoq.com/br/presentations/deep-recommender-systems/

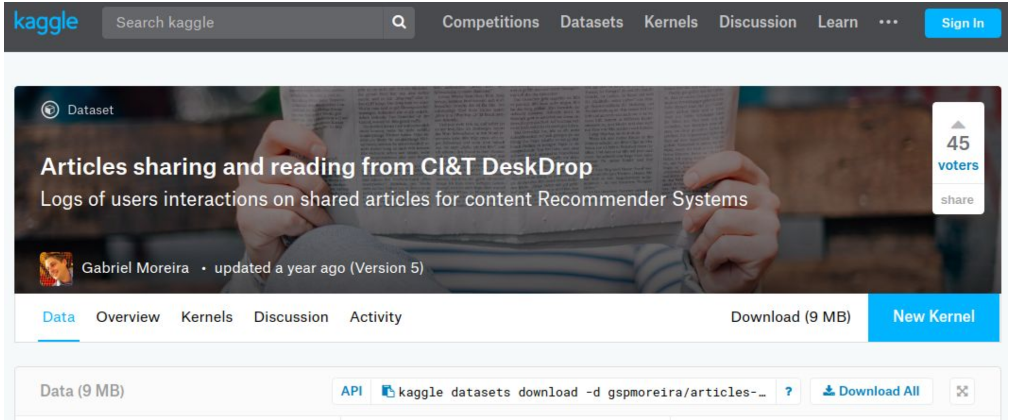
References


Recommender Systems
An Introduction
DIETMAR JANNACH
MARKUS ZANKER
ALEXANDER FELFERNIG
GERHARD FRIEDRICH
CAMBRIDGE


Copyrighted Material
FRANCESCO RICCI
LIOR ROKACH
BRACHA SHAPIRA
PAUL B. KANTOR EDITORS
RECOMMENDER SYSTEMS HANDBOOK
Springer
Copyrighted Material


Charu C. Aggarwal
Recommender Systems
The Textbook
Springer

CI&T Deskdrop dataset on Kaggle!



- 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
[@gspmoreira]

CI&T