Third Workshop on Deep Learning for Recommender Systems (DLRS)
ACM RecSys 2018

# News session-based recommendations using Deep Neural Networks

Gabriel Moreira (CI&T | ITA)    Felipe Ferreira (globo.com)    Adilson Cunha (ITA)

01

# Introduction

## 01.About Me

# Felipe Ferreira

- PhD student - Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
- Master Degree in Computer Science - Federal University of Amazonas (UFAM)
- Machine Learning Engineer - Globo.com

02

# About Globo.com

globo
.com

# Leader in audience and one of the main technology companies in Brazil

OUR MAIN OFFICE IS IN

# Rio de Janeiro

Other offices: São Paulo and Porto Alegre globo.com

# 87 MILLION

UNIQUE USER'S PER MONTH

comScore feb/2018

**4 billion** daily events

**10 million** unique users per day

**2 million** concurrent connections

**100 thousand** new content per month

03

# News Challenges
for Recommender Systems

# 1. Sparse user profiling

Majority of readers are anonymous (no past information) and read a few stories from the entire repository

# 2. Fast growing number of items

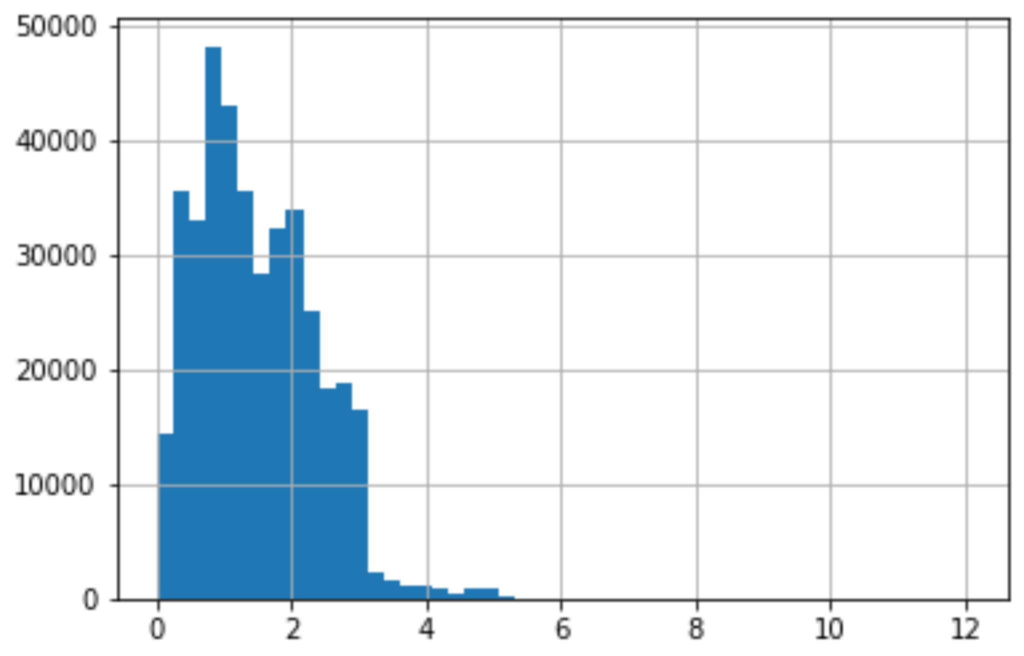
Thousands of new stories added daily in news portals

# 3. User preferences shift

News topics of interests are not as stable as in the entertainment domain. Users short-term and long-term interests influence whether to read an article

# 4. Accelerated decay of item's value

Most users are interested in fresh information and news articles are expected to have a short shelf life (e.g. few day or hours)



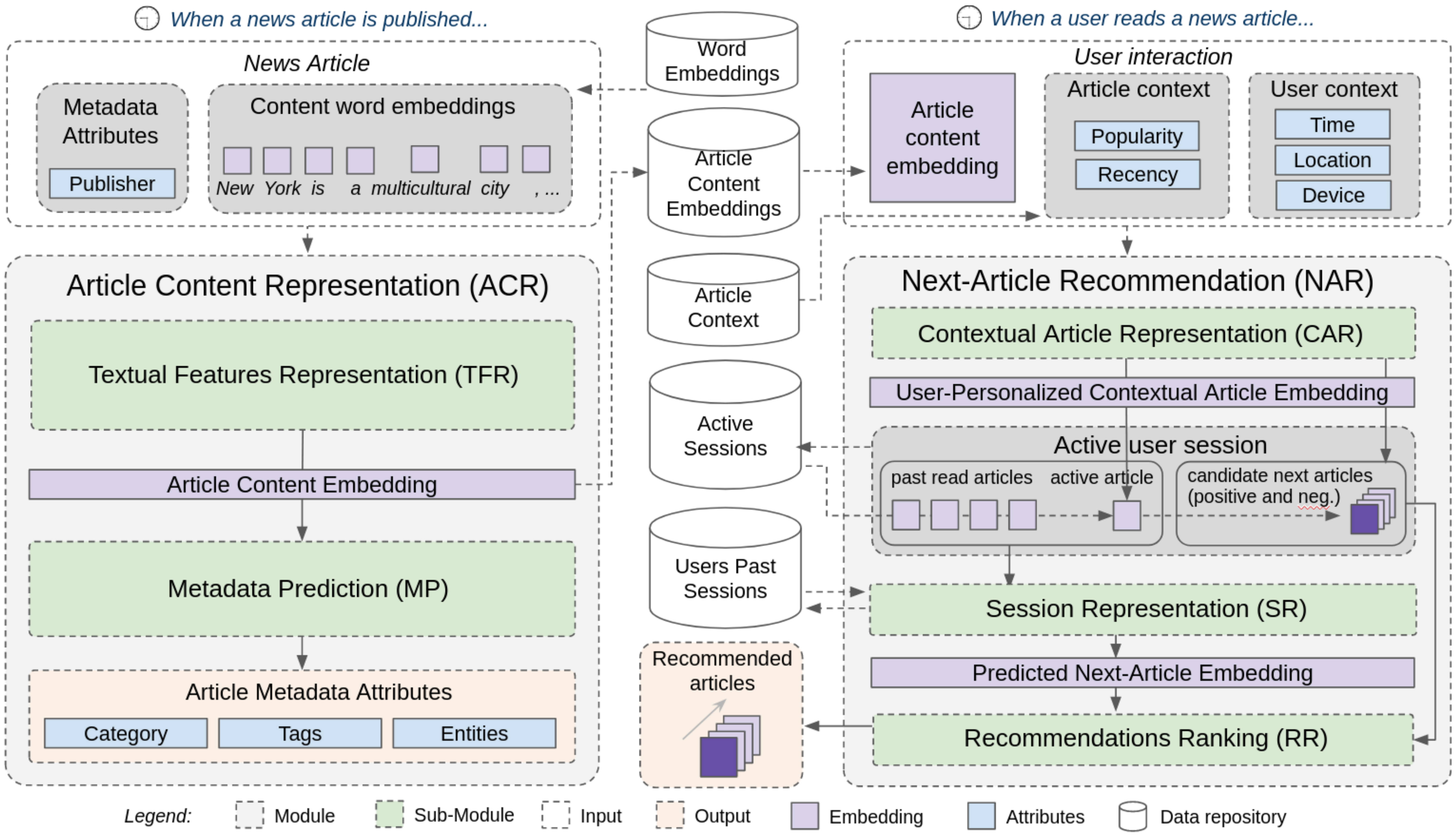
04

# CHAMELEON

## a Deep-Learning Meta-Architecture for News Recommender Systems

# CHAMELEON

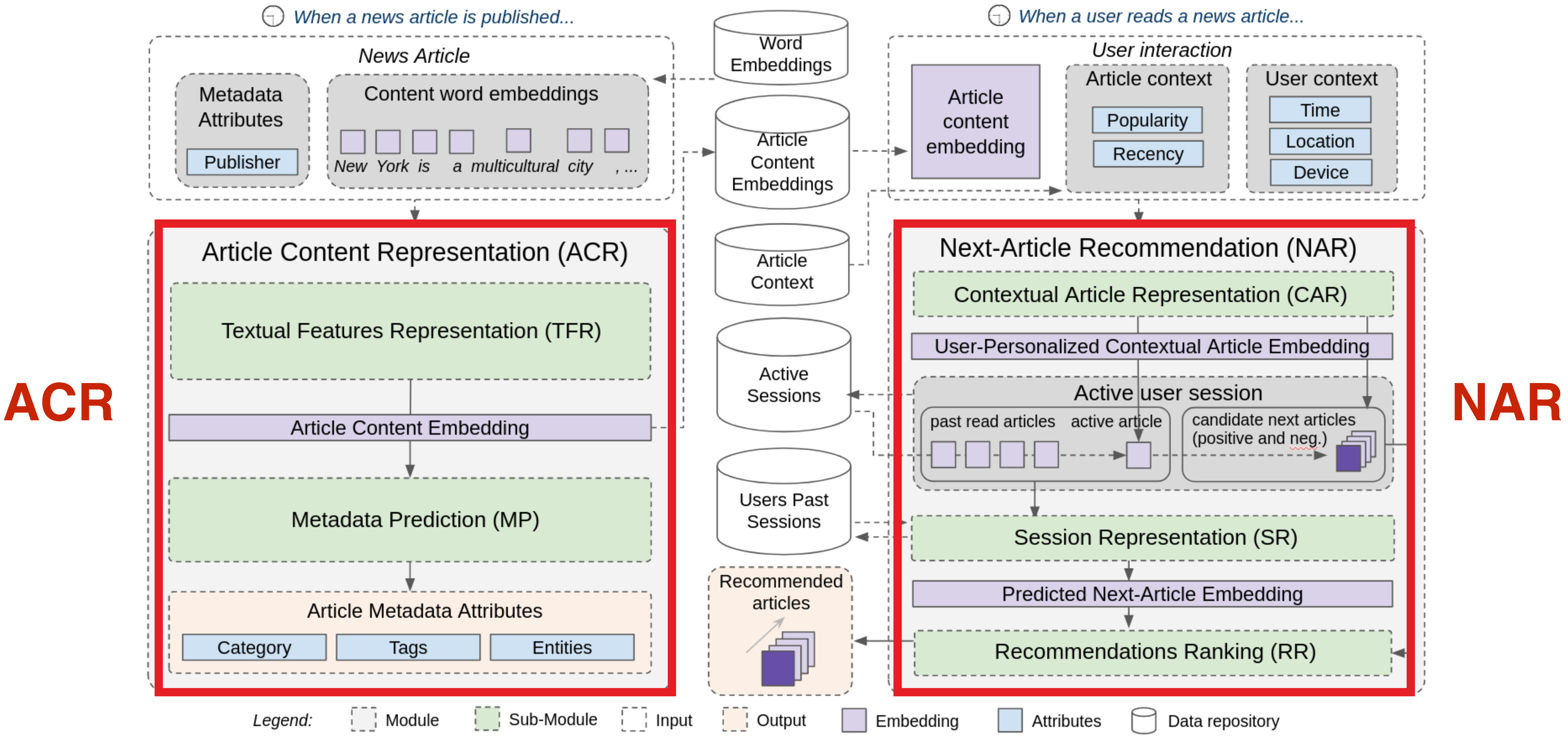
CHAMELEON is composed of two complementary modules




Moreira, 2018

# CHAMELEON

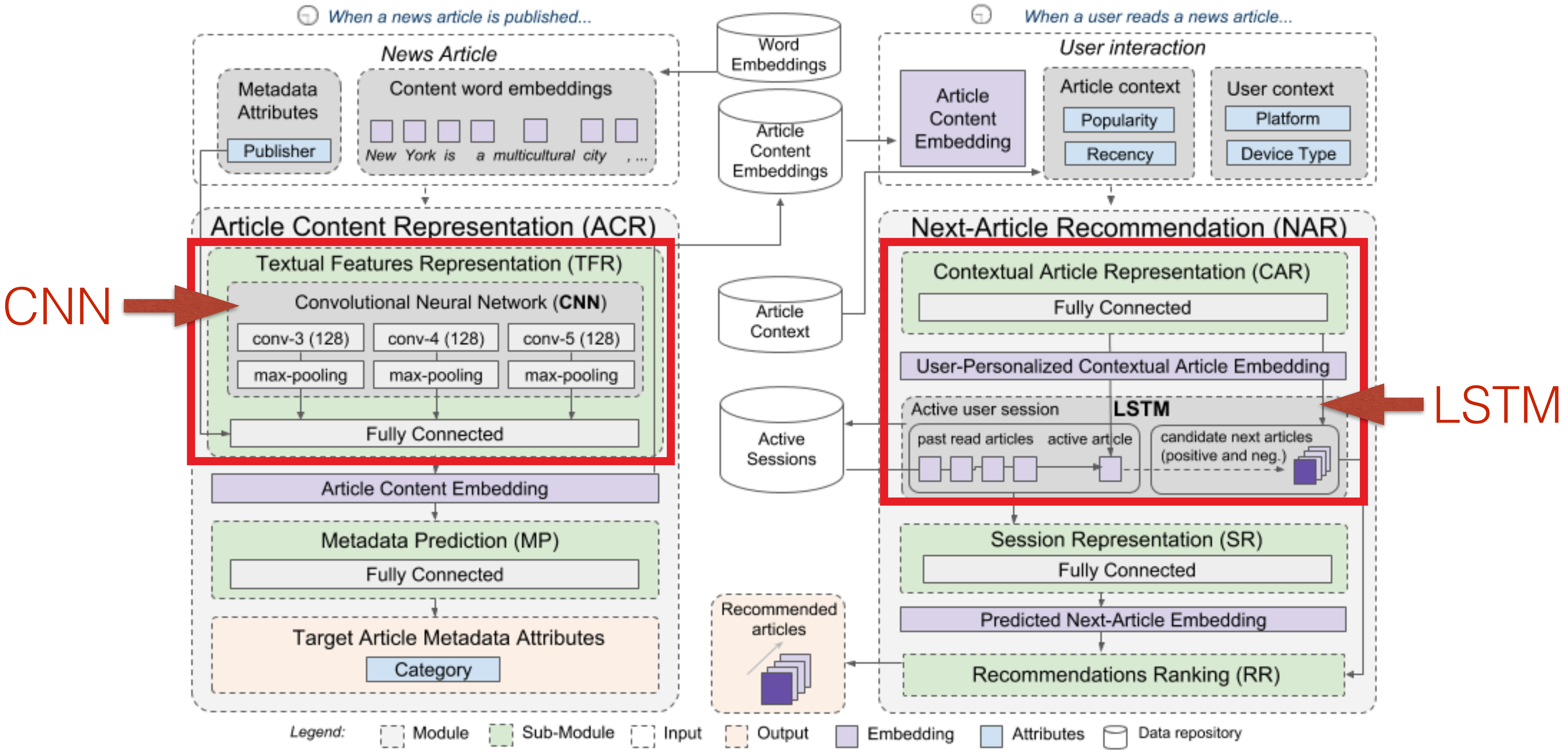
CHAMELEON is composed of two complementary modules




Moreira, 2018

# CHAMELEON

An architecture instantiation

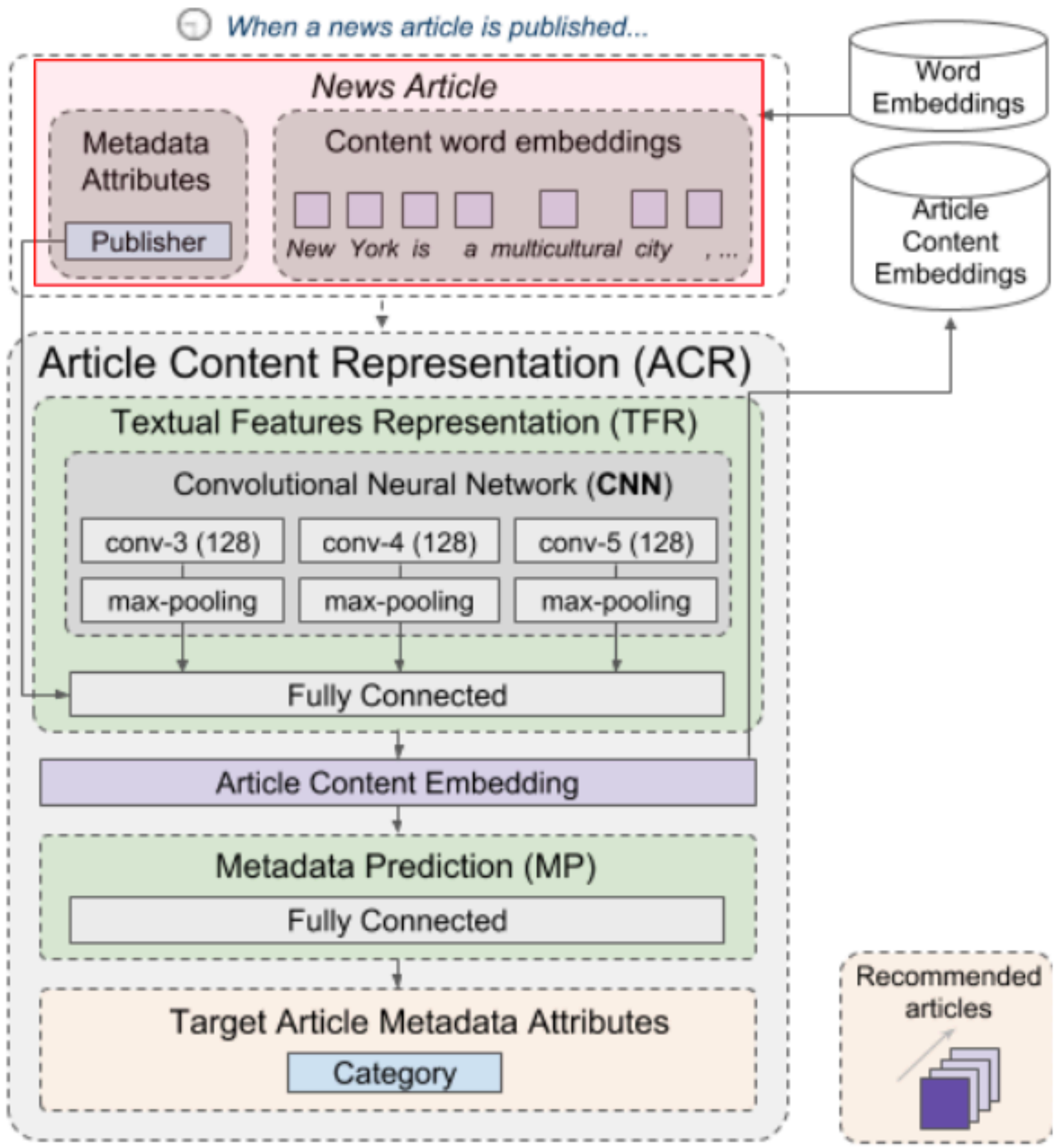



Moreira, 2018

# ACR Module

# ACR Module

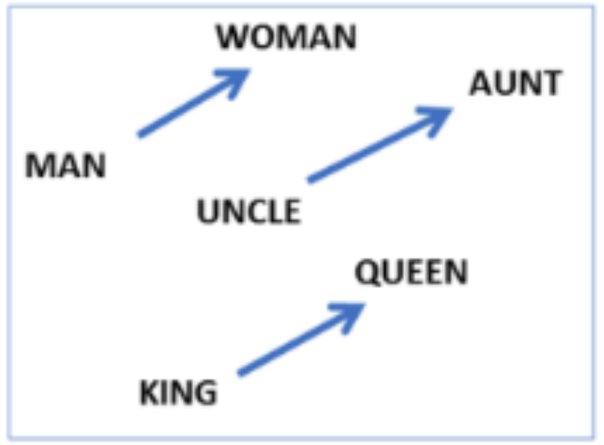
part 1 of 4




## Inputs

**Article text represented by Word embeddings**

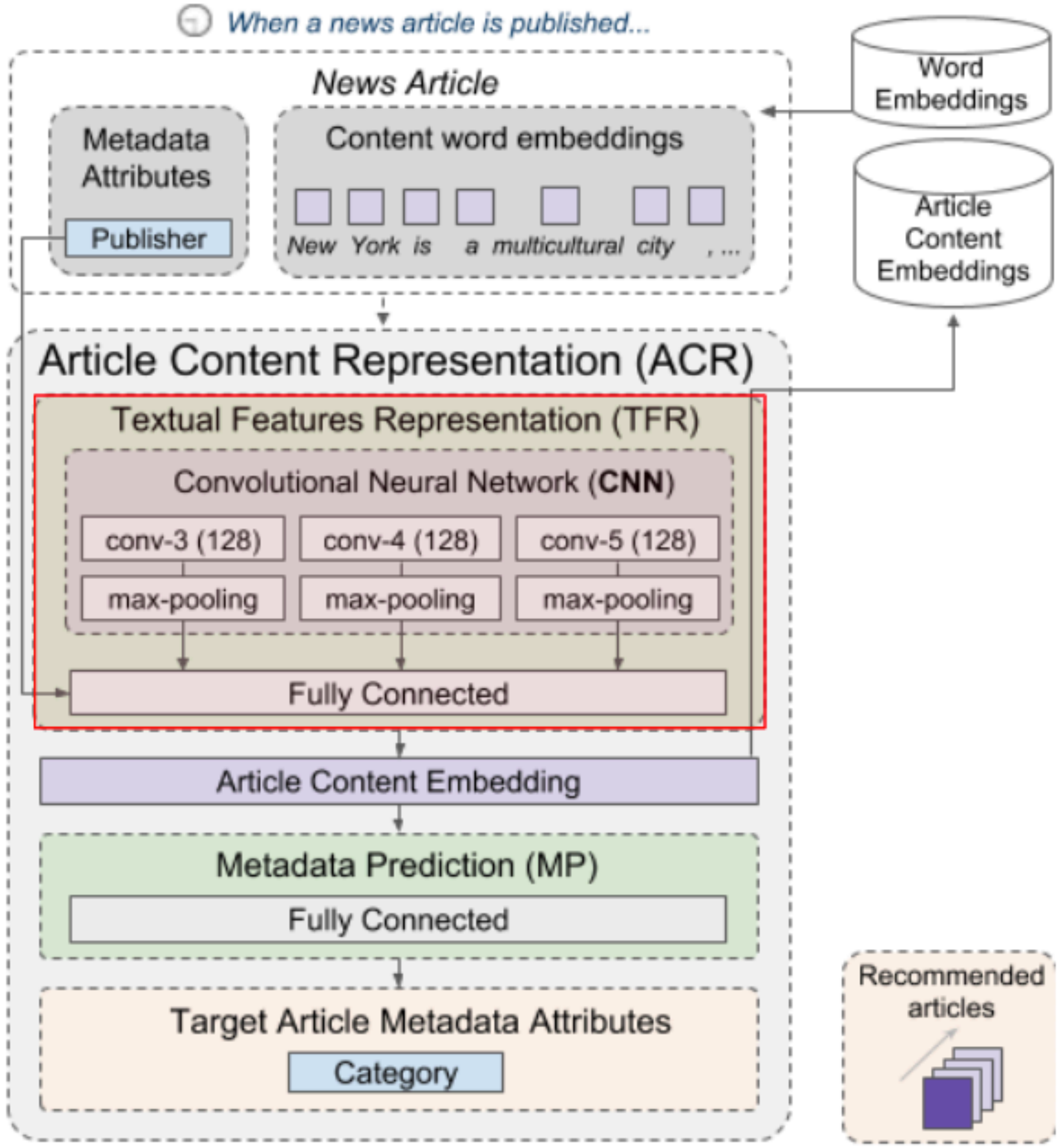
Pre-trained Portuguese word embeddings (Word2Vec)




**Article metadata attributes** (Publisher)

# ACR Module

part 2 of 4

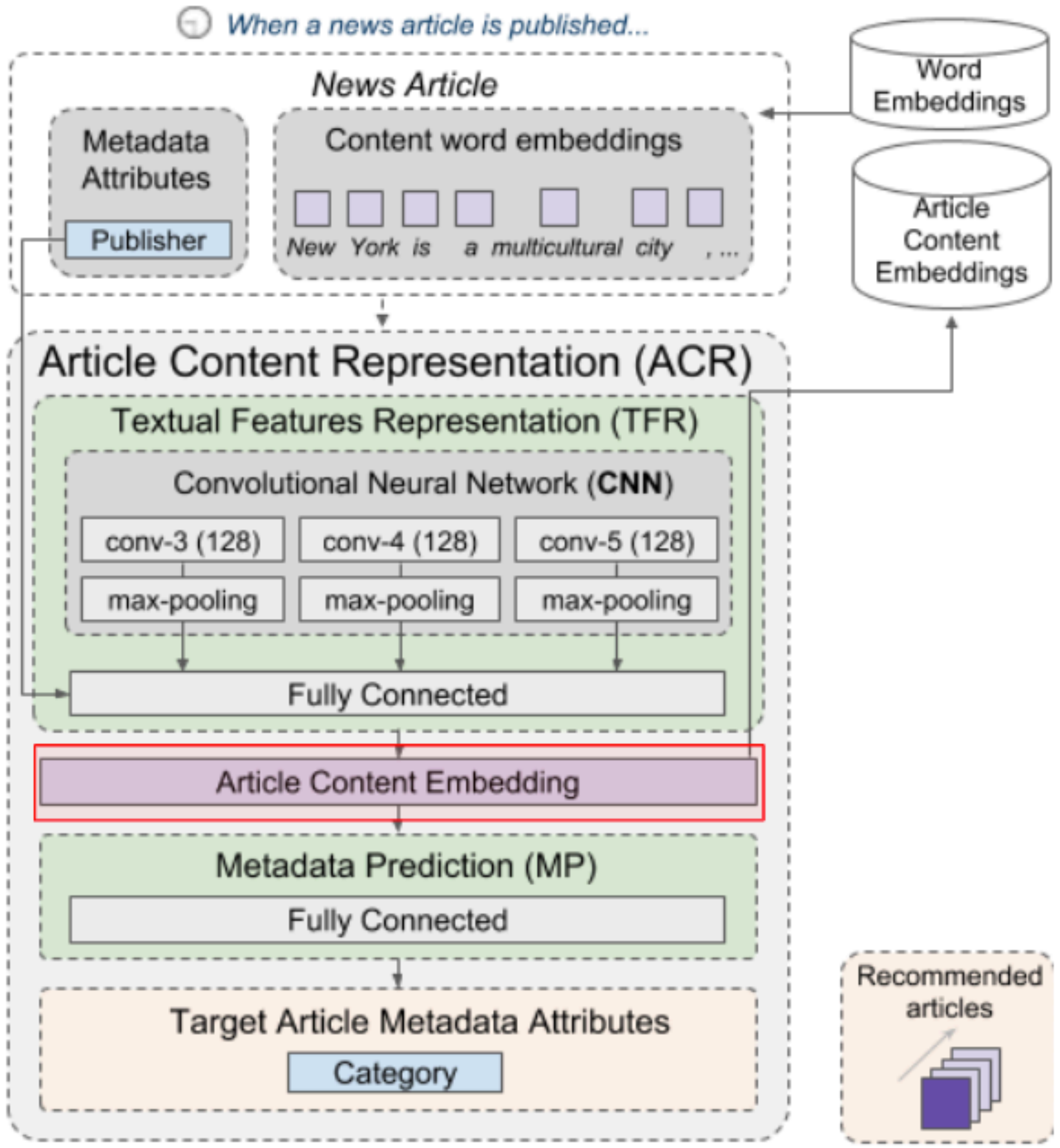



## Textual Feature Representation

- 1D CNN layers with window sizes of 3, 4, and 5 words and 128 filters each.
- A Fully Connected layer to combine the CNN feature maps and metadata attributes.

# ACR Module

part 3 of 4

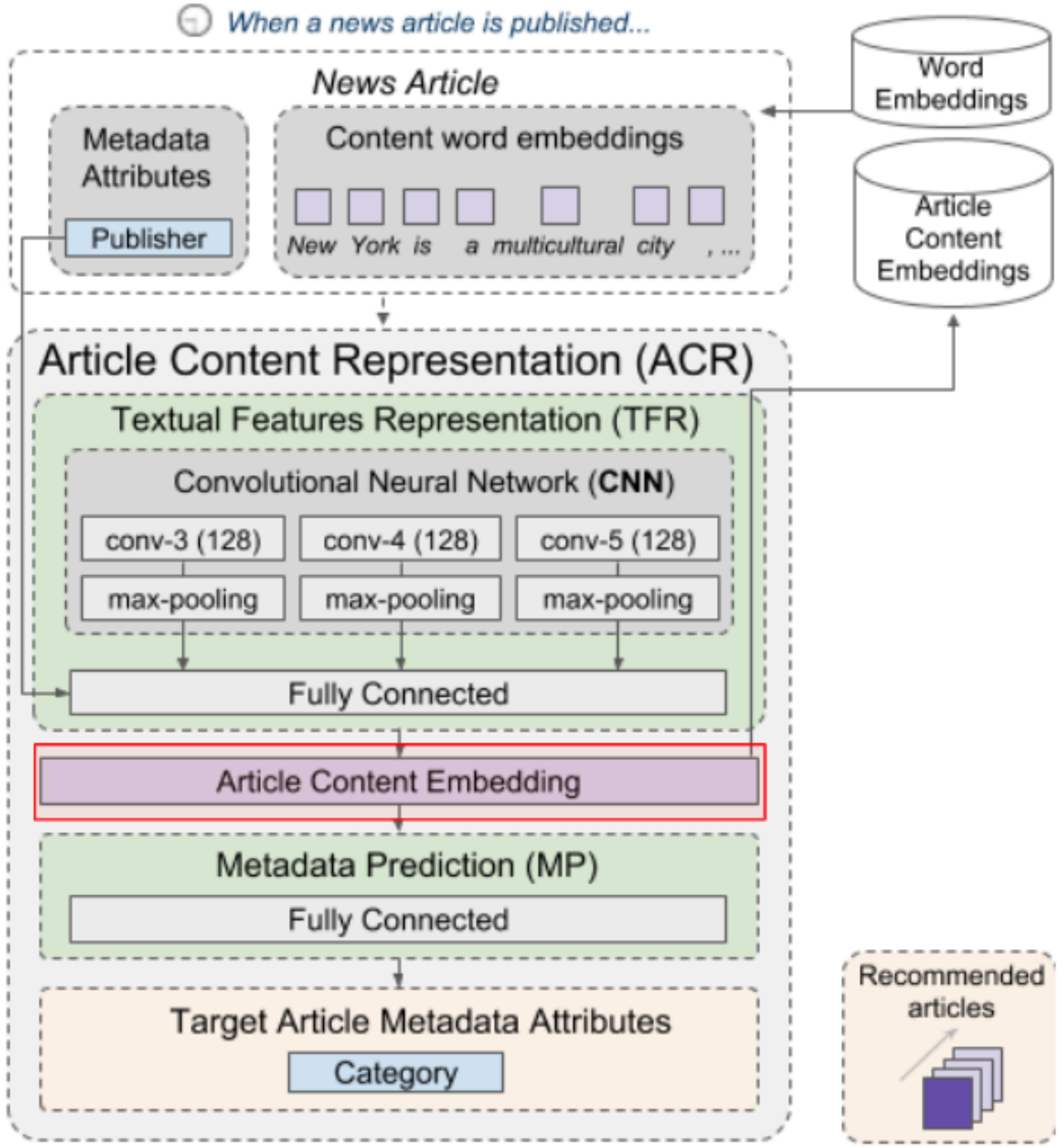



### Article Content Embedding

- The learned article distributed representation
- After training, it is stored in a repository for further usage by NAR module

# ACR Module

part 3 of 4

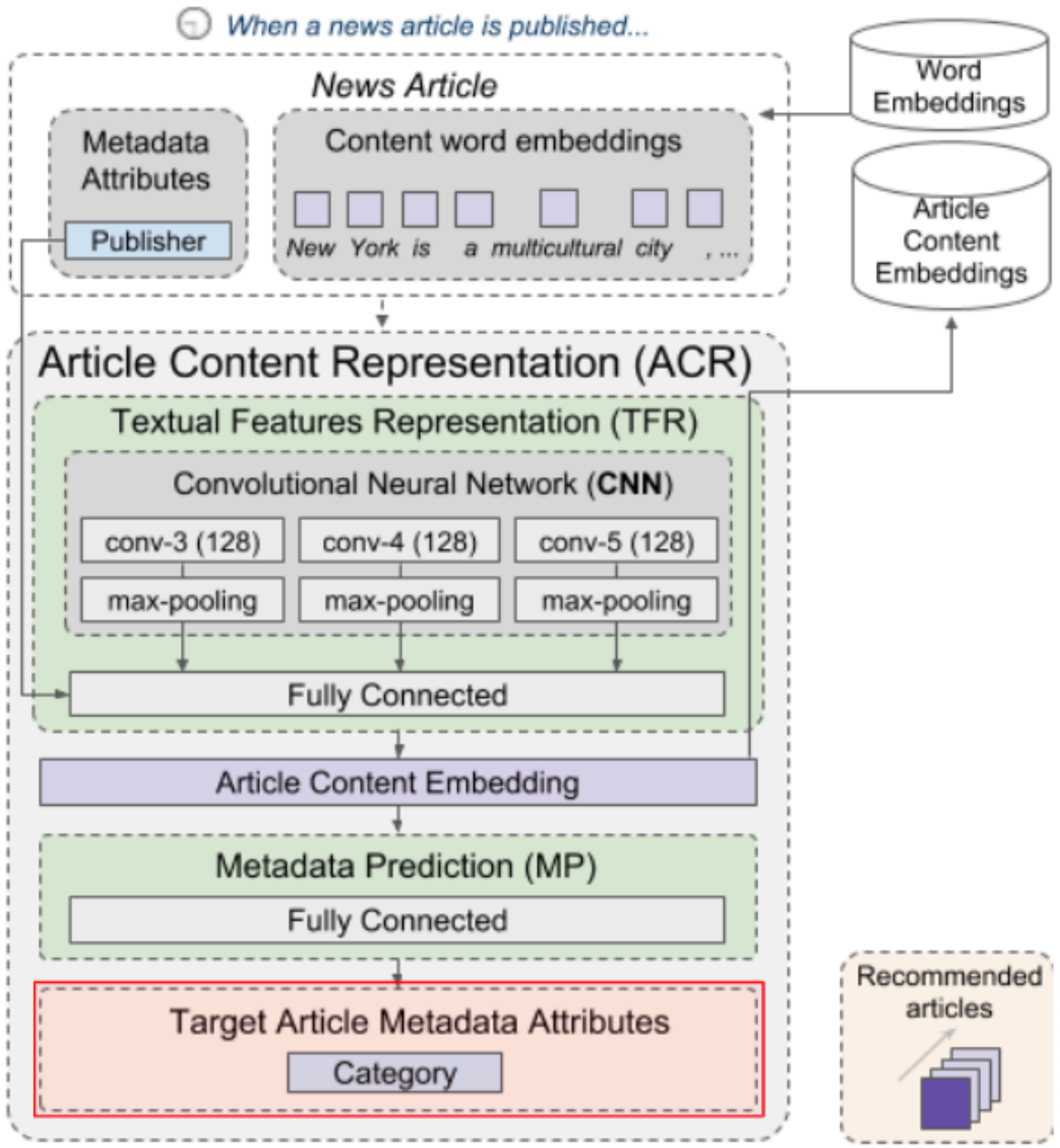



## Article Content Embedding

- The learned article distributed representation
- After training, it is stored in a repository for further usage by NAR module

# ACR Module

part 4 of 4




**Output: Article Metadata Attributes**

The last layer of ACR module is responsible to classify one or more article metadata attributes

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

*Eq. 1 - Softmax*

$$l(\theta) = -\frac{1}{N}(\sum_{i=1}^{N} y_i \cdot log(\hat{y}_i)) + \lambda\|\theta\|$$

*Eq. 2 - Cross-entropy*

# NAR Module

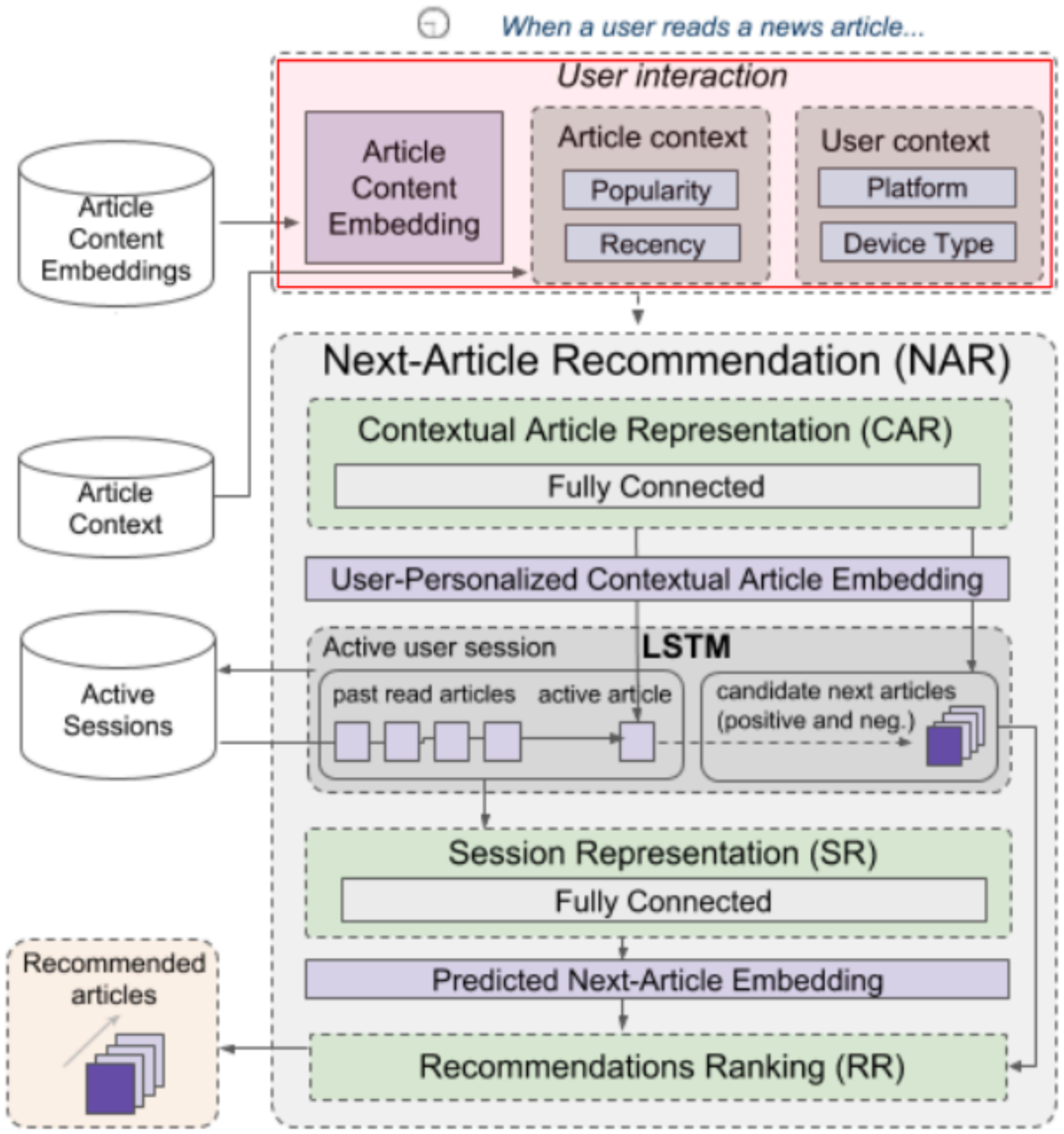
# NAR Module

part 1 of 6

- Predicts the next-clicked item for users active sessions.
- Sessions are sequences of user interactions with no more than 30 minutes between them.
- Sessions with less than 2 interactions or more than 20 interactions were discarded.

Sessions mini-batch

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



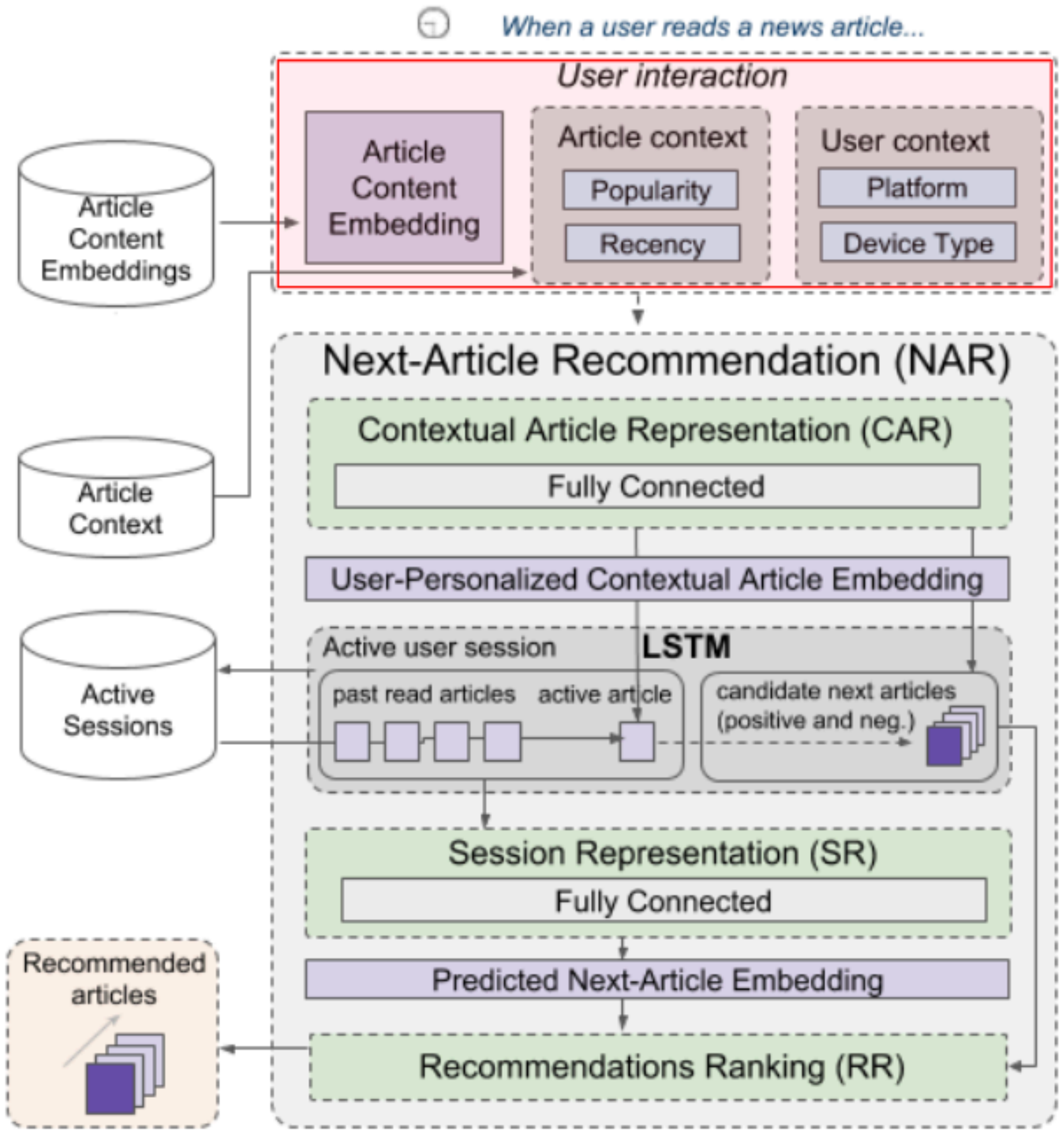
# NAR Module

part 1 of 6

- Predicts the next-clicked item for users active sessions.
- Sessions are sequences of user interactions with no more than 30 minutes between them.
- Sessions with less than 2 interactions or more than 20 interactions were discarded.

Sessions mini-batch

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



# NAR Module

part 2 of 6

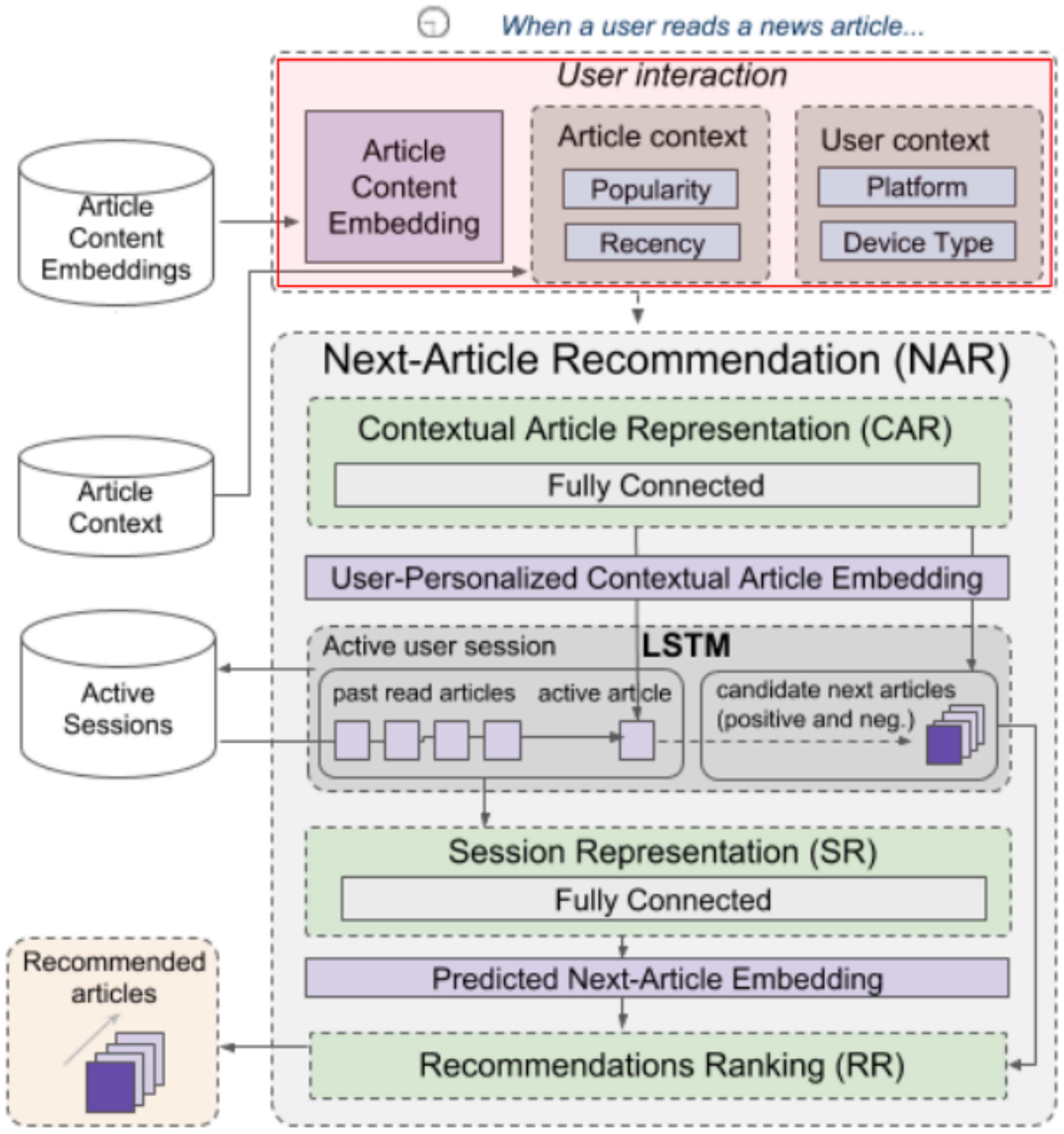
**Inputs**

**Article Content Embeddings**
Pre-trained by the ACR module

**User Context**

- Platform
- Device time

Could also include Time (e.g., hour of the day, weekday/ weekend) and Location contextual information (e.g. city, lat/long)



# NAR Module

part 2 of 6

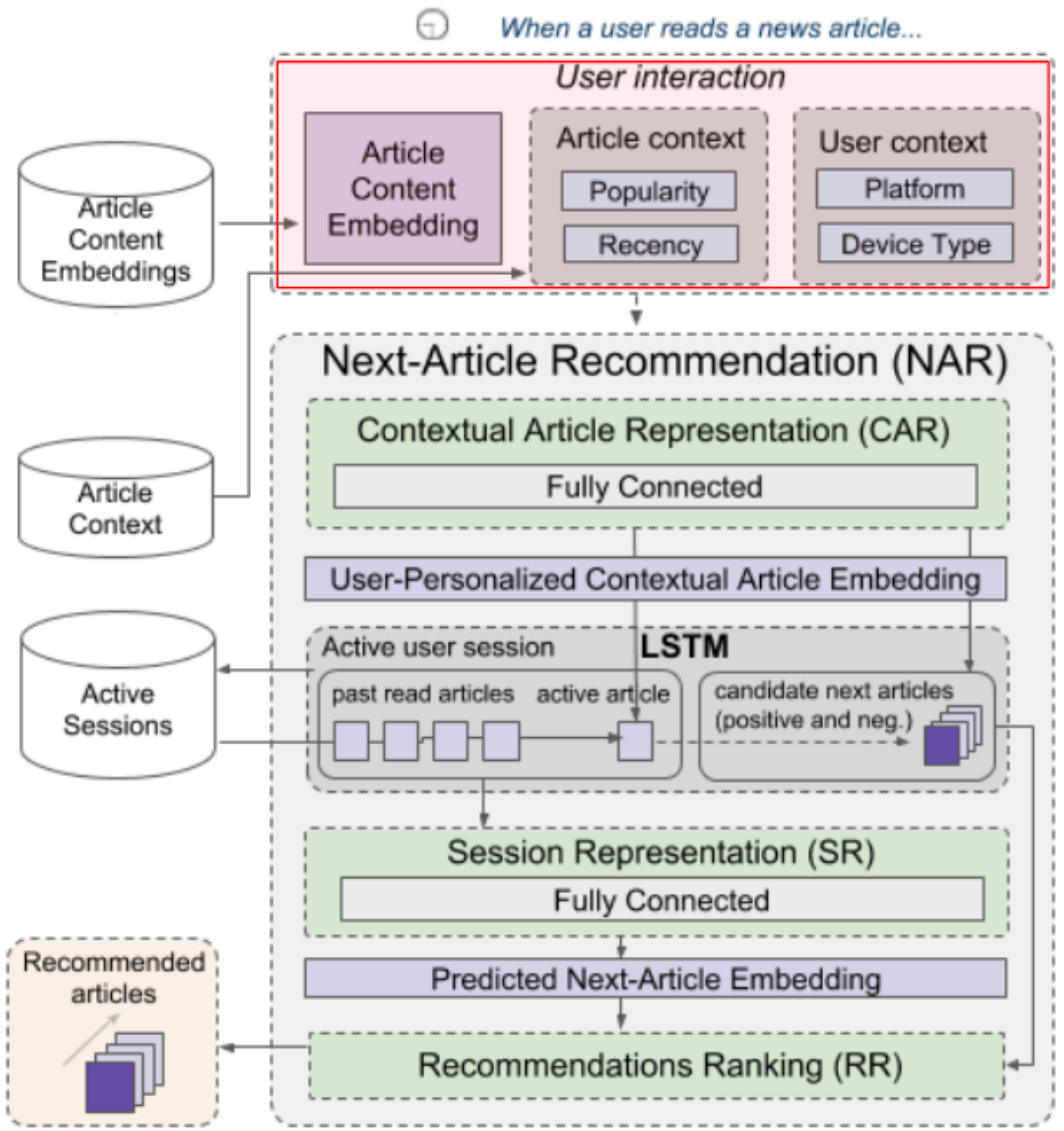
## Inputs

### Article Context

- Recent Popularity
- Recency

### Articles Context temporal update method:

1. Keep a global buffer with the last N clicks/hours (article reads)
2. Compute articles recent popularity by counting their clicks within the buffer
3. Compute articles recency as the number of elapsed hours since article was published
4. For each article read by a user, look up for the updated article context features: (recent popularity and recency)

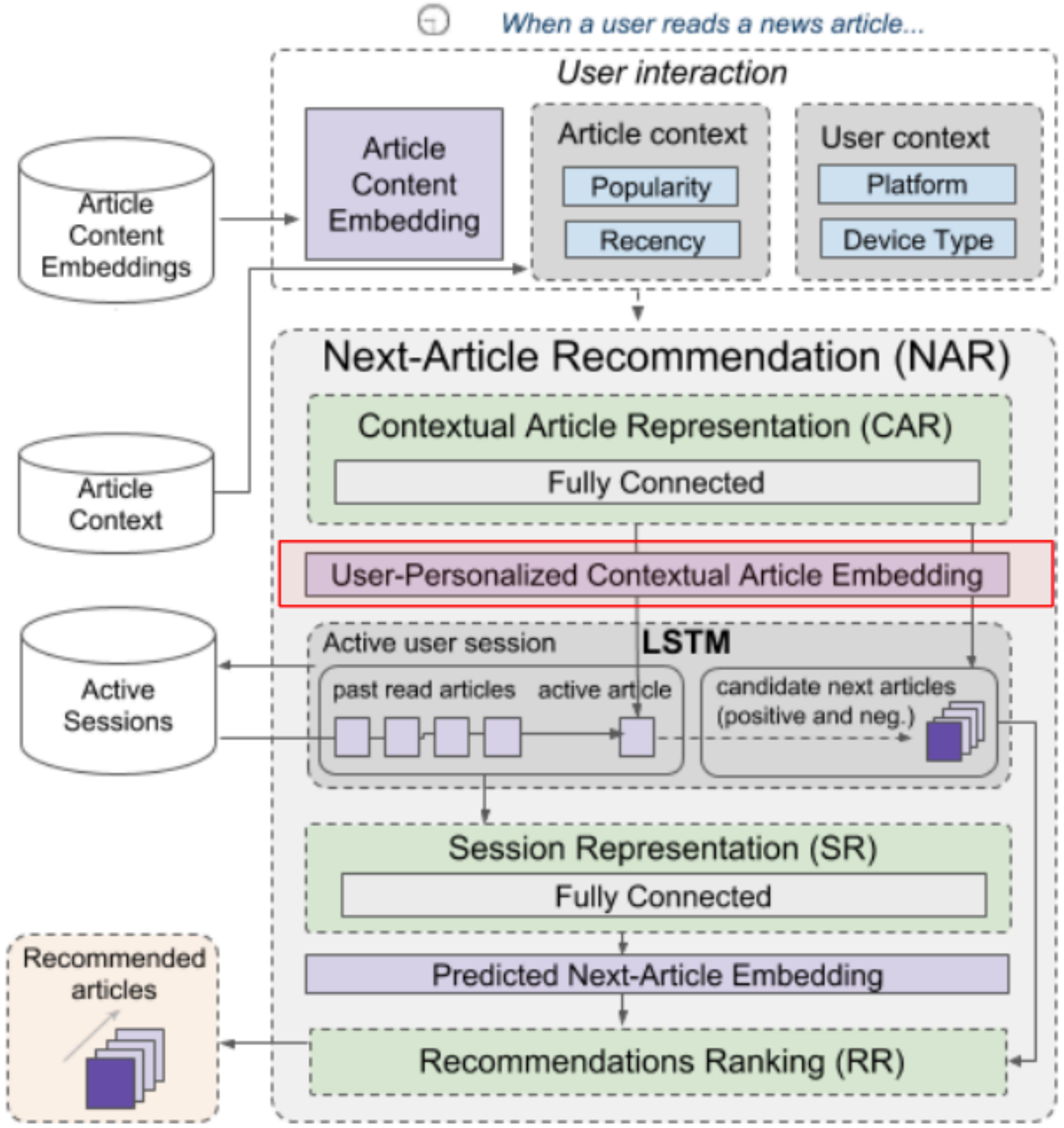


# NAR Module

part 3 of 6

## User-Personalized Contextual Article Embedding

Merges inputs to create a representation that combines the article content, article context and user context.



# NAR Module

part 4 of 6

### Active User Session

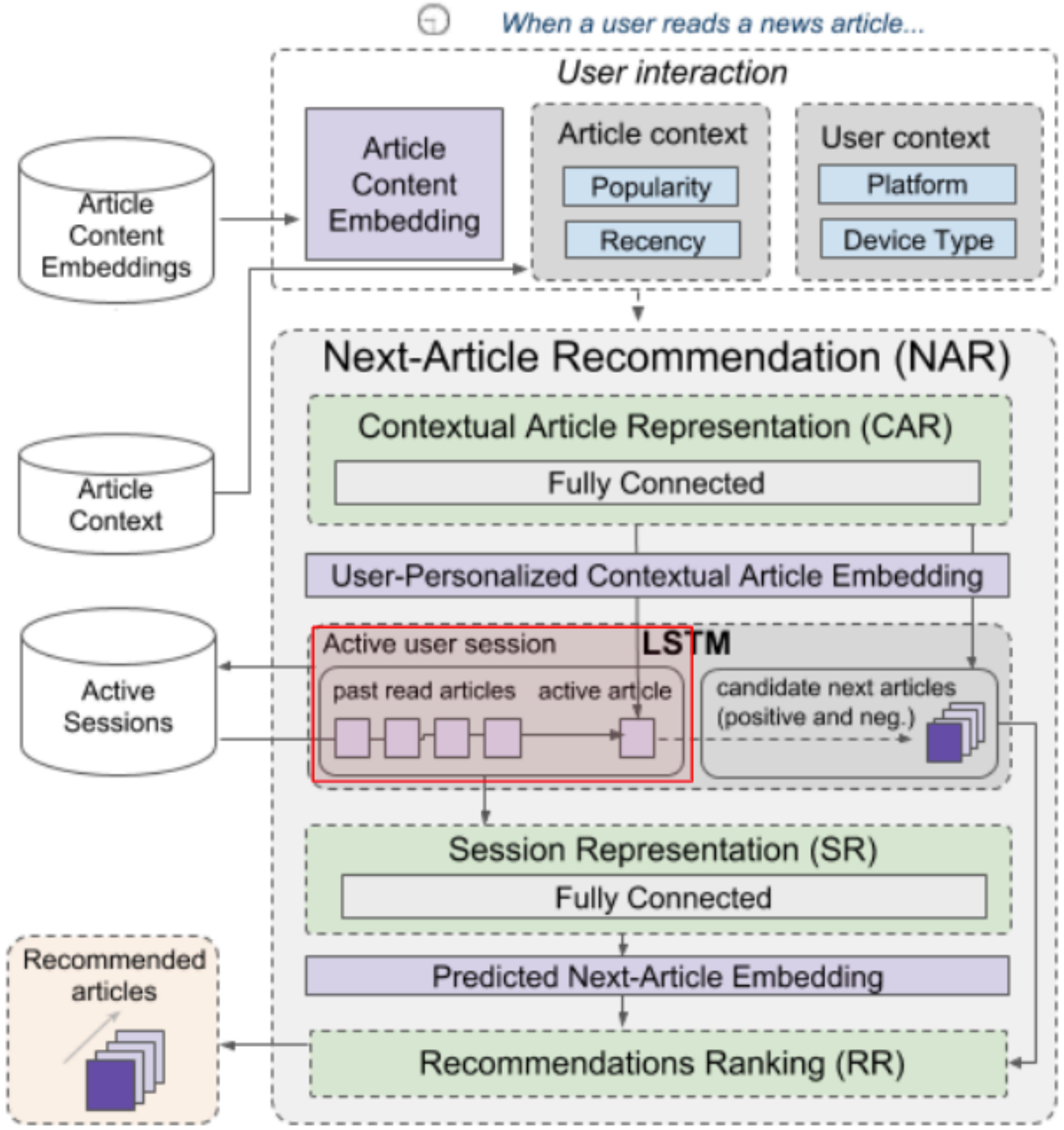
Trains an RNN (LSTM) to predict next-clicked items (represented by their *User-Personalized Contextual Article Embedding*), for each active session:

**RNN input**

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

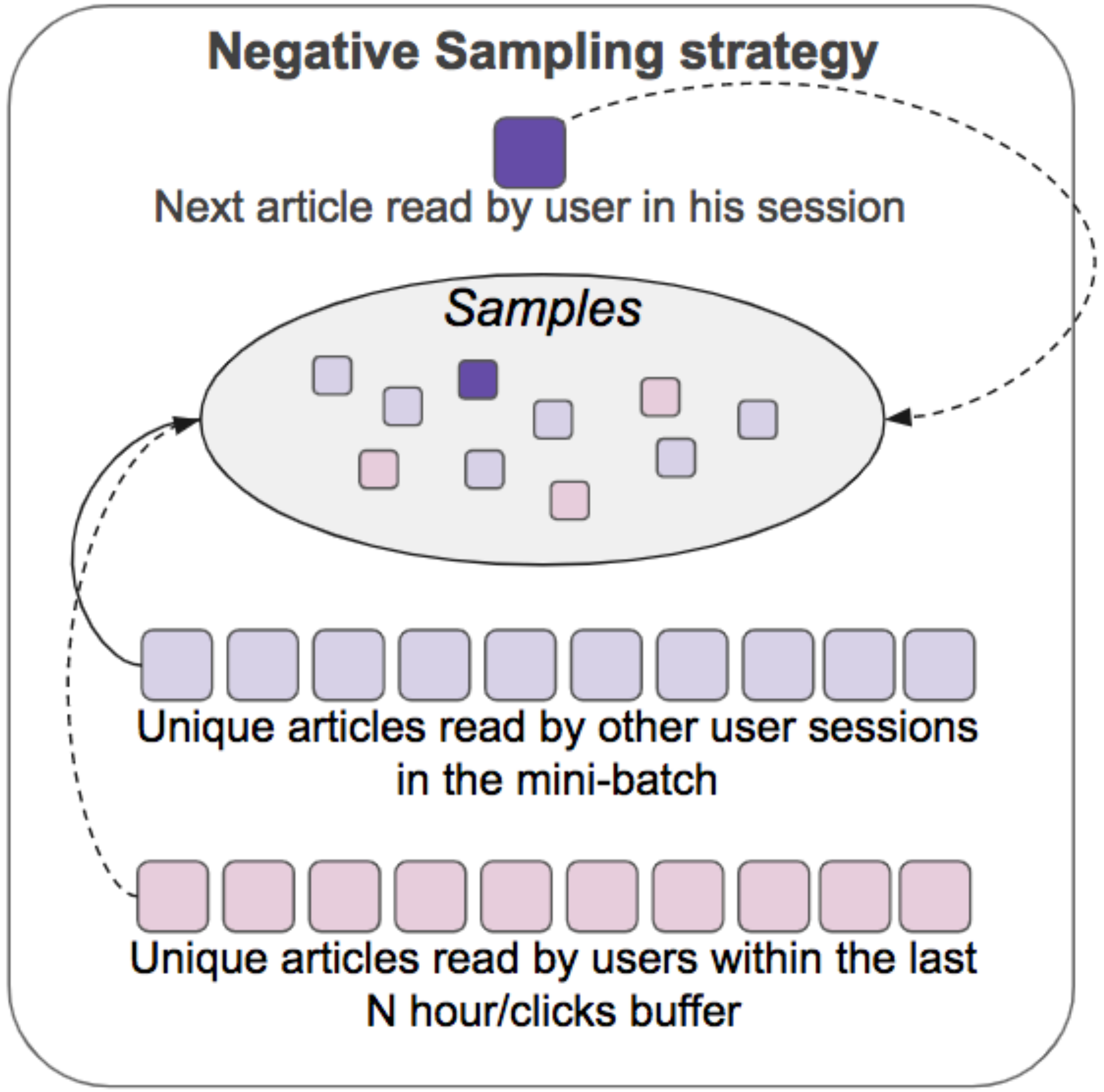
**Expected RNN output (next-clicked items)**

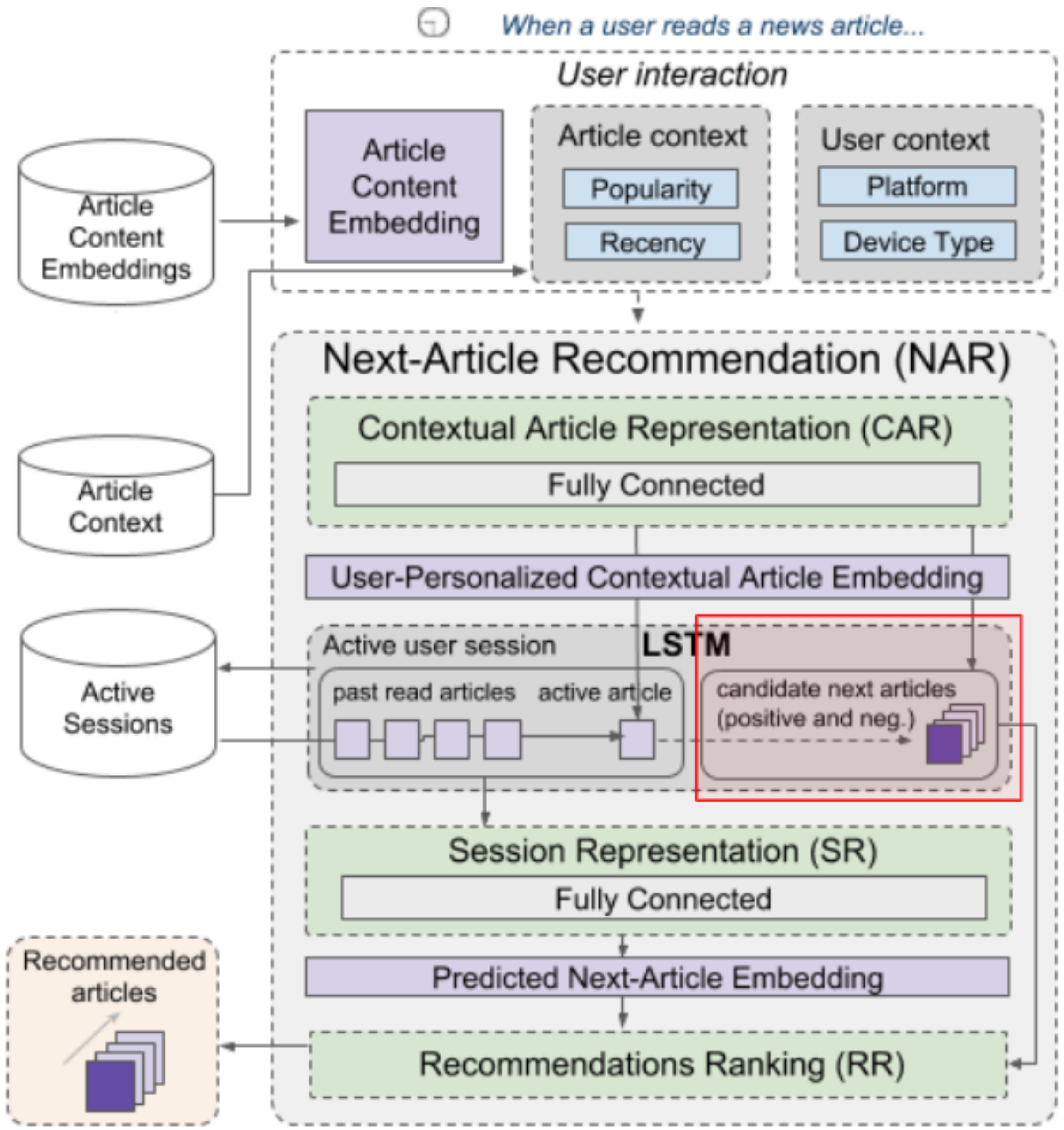
| | | | |
|---|---|---|---|
| $I_{1,2}$ | $I_{1,3}$ | $I_{1,4}$ | $I_{1,5}$ |



# NAR Module

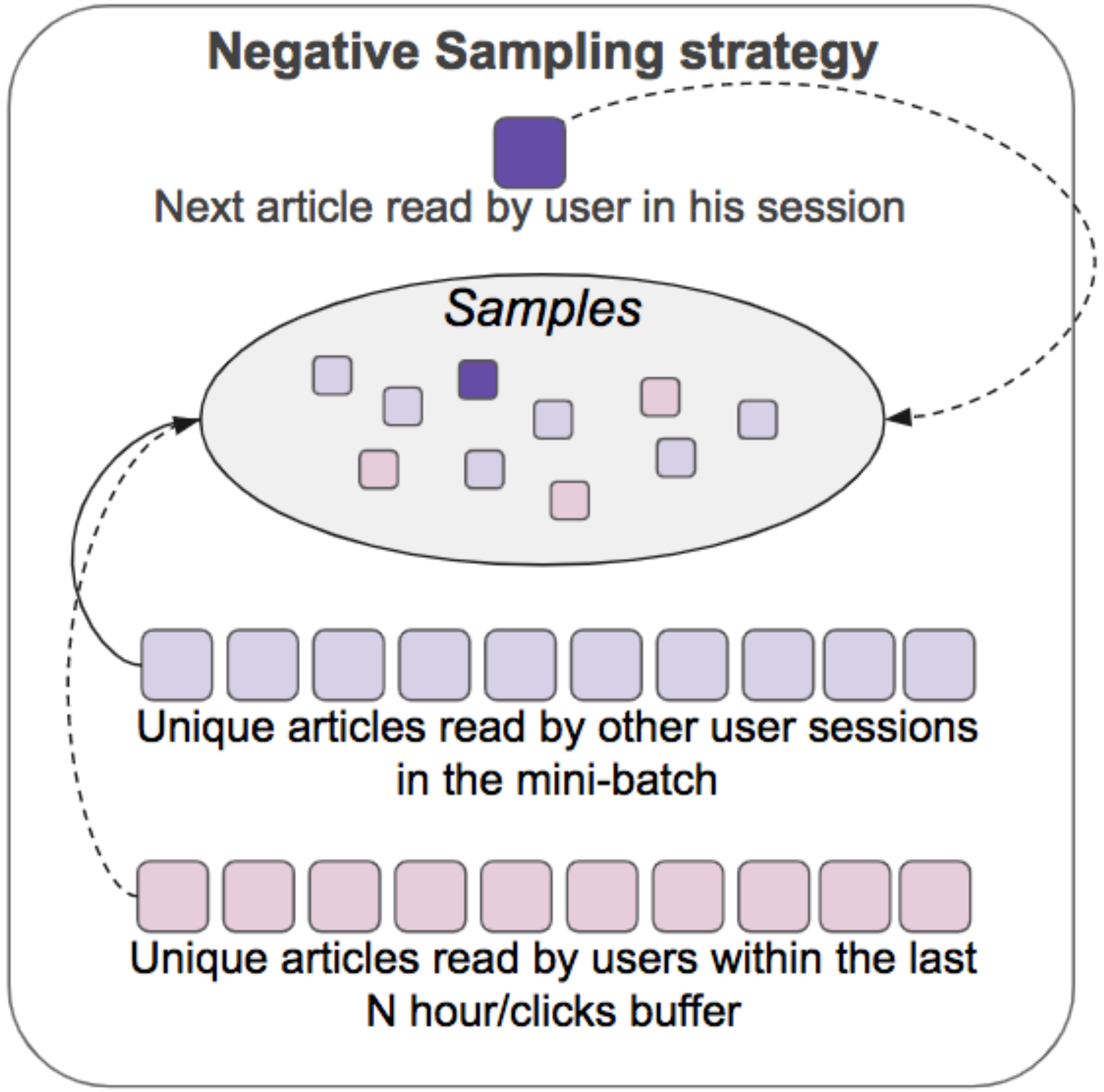
part 5 of 6

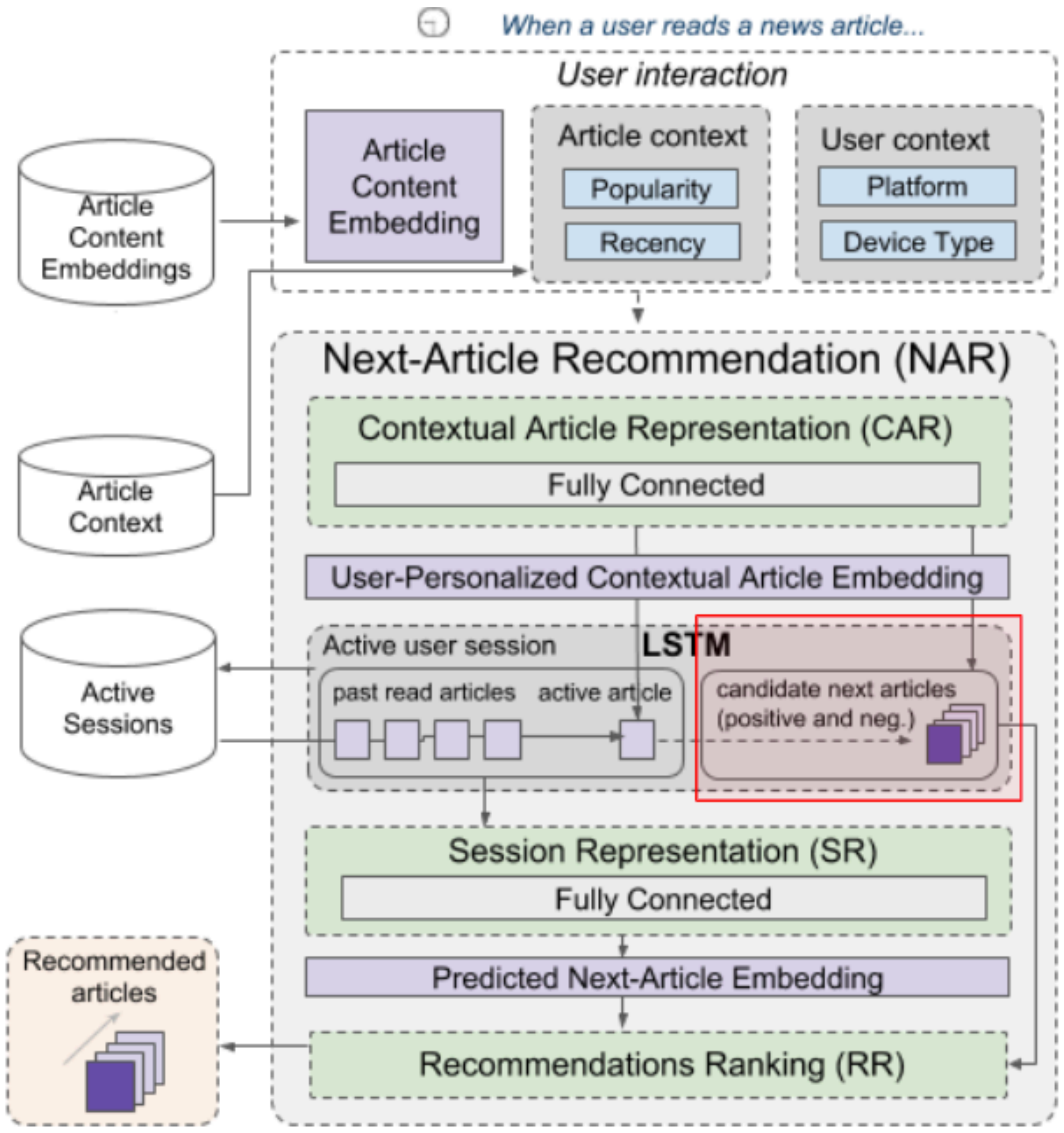


# NAR Module

part 5 of 6



# NAR Module

part 6 of 6

## Recommendations Ranking (RR) sub-module

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

*Eq. 3 - Relevance Score of an item for a user session*

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

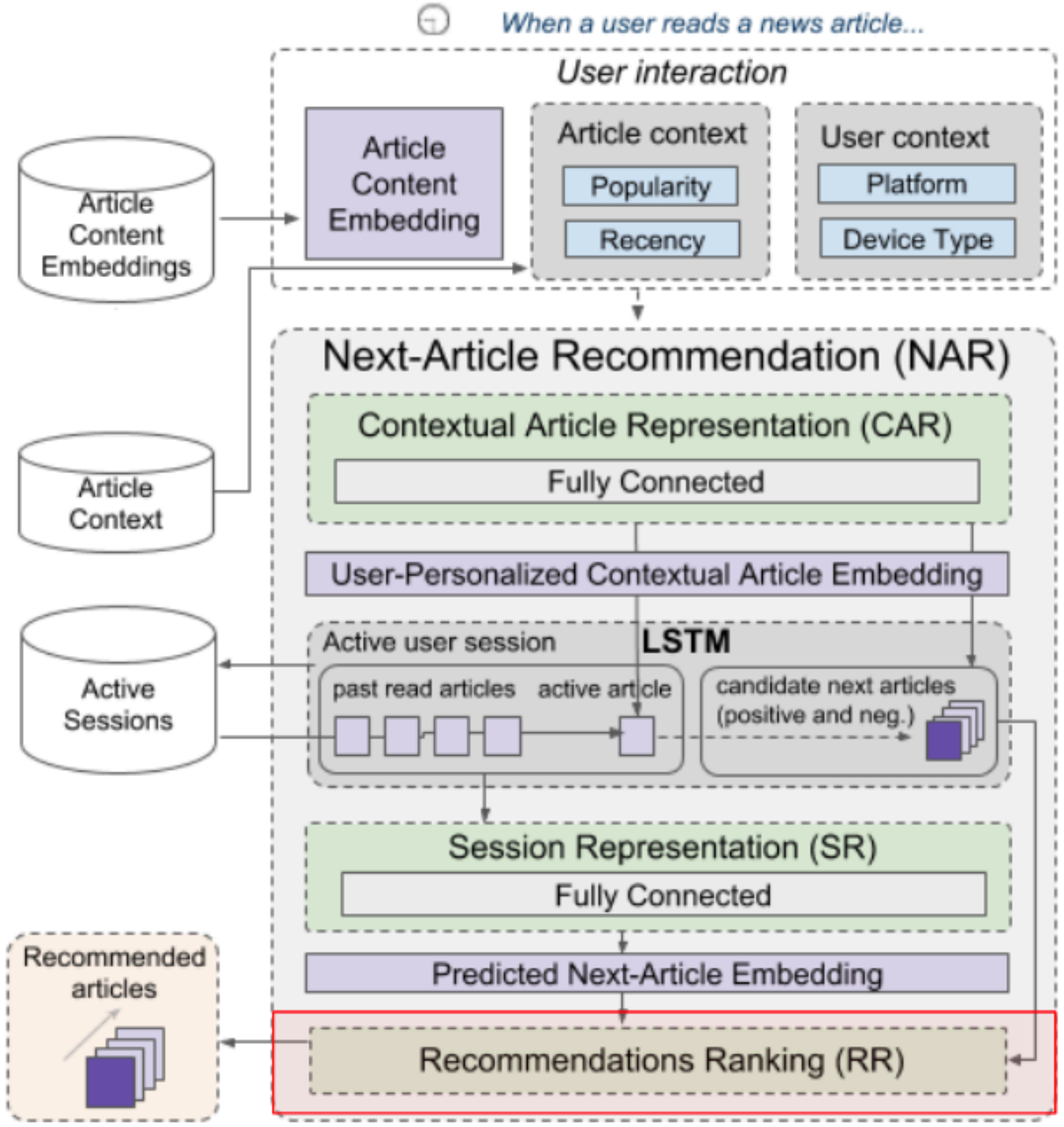
*Eq. 4 - Cosine similarity*

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

*Eq. 5 - Softmax over Relevance Score (HUANG et al., 2013)*

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

*Eq. 6 - Loss function (HUANG et al., 2013)*



# NAR Module

part 6 of 6

**Recommendations Ranking (RR) sub-module**

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

*Eq. 3 - Relevance Score of an item for a user session*

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

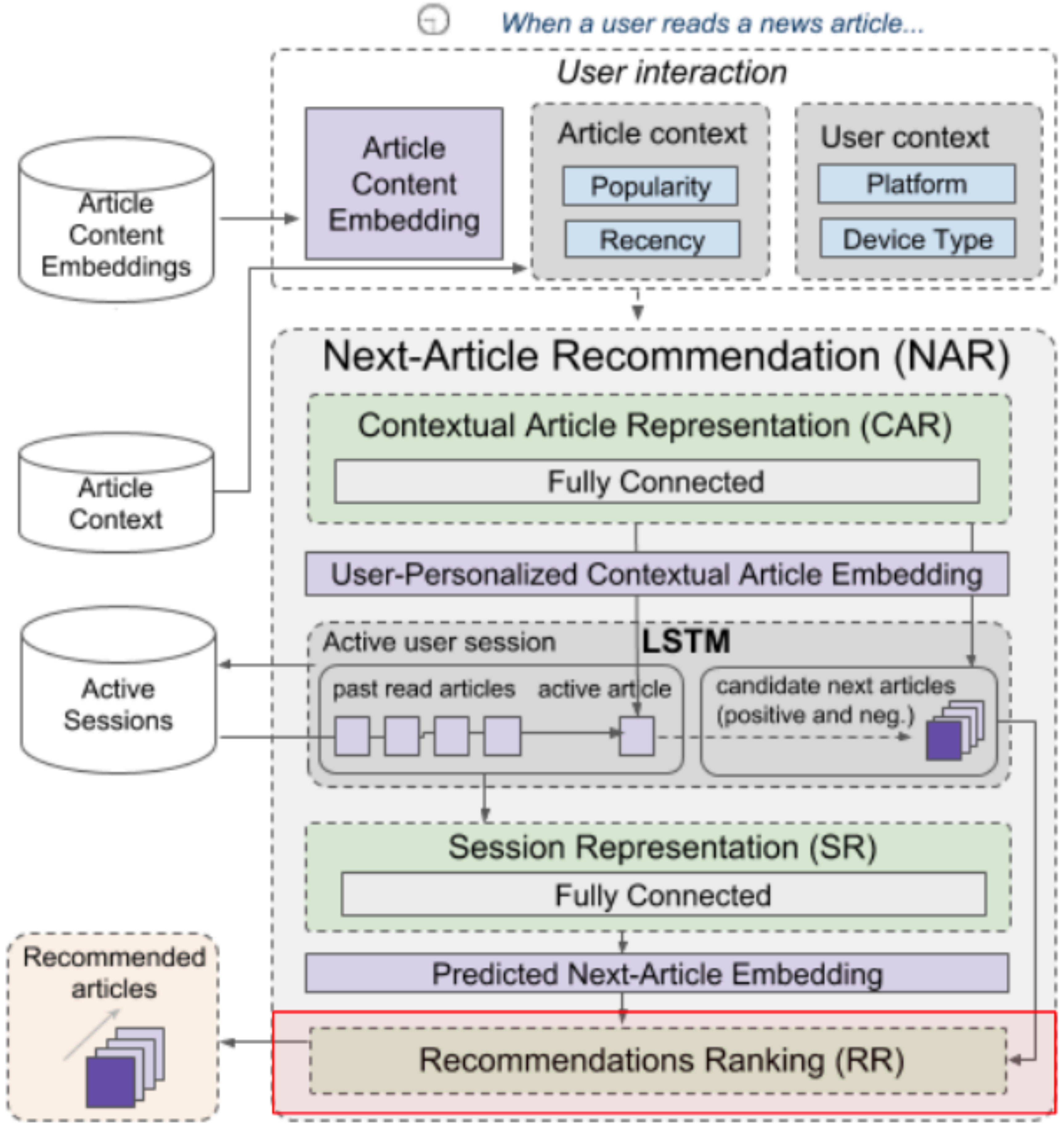
*Eq. 4 - Cosine similarity*

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

*Eq. 5 - Softmax over Relevance Score (HUANG et al., 2013)*

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

*Eq. 6 - Loss function (HUANG et al., 2013)*



# NAR Module

part 6 of 6

## Recommendations Ranking (RR) sub-module

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

*Eq. 3 - Relevance Score of an item for a user session*

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

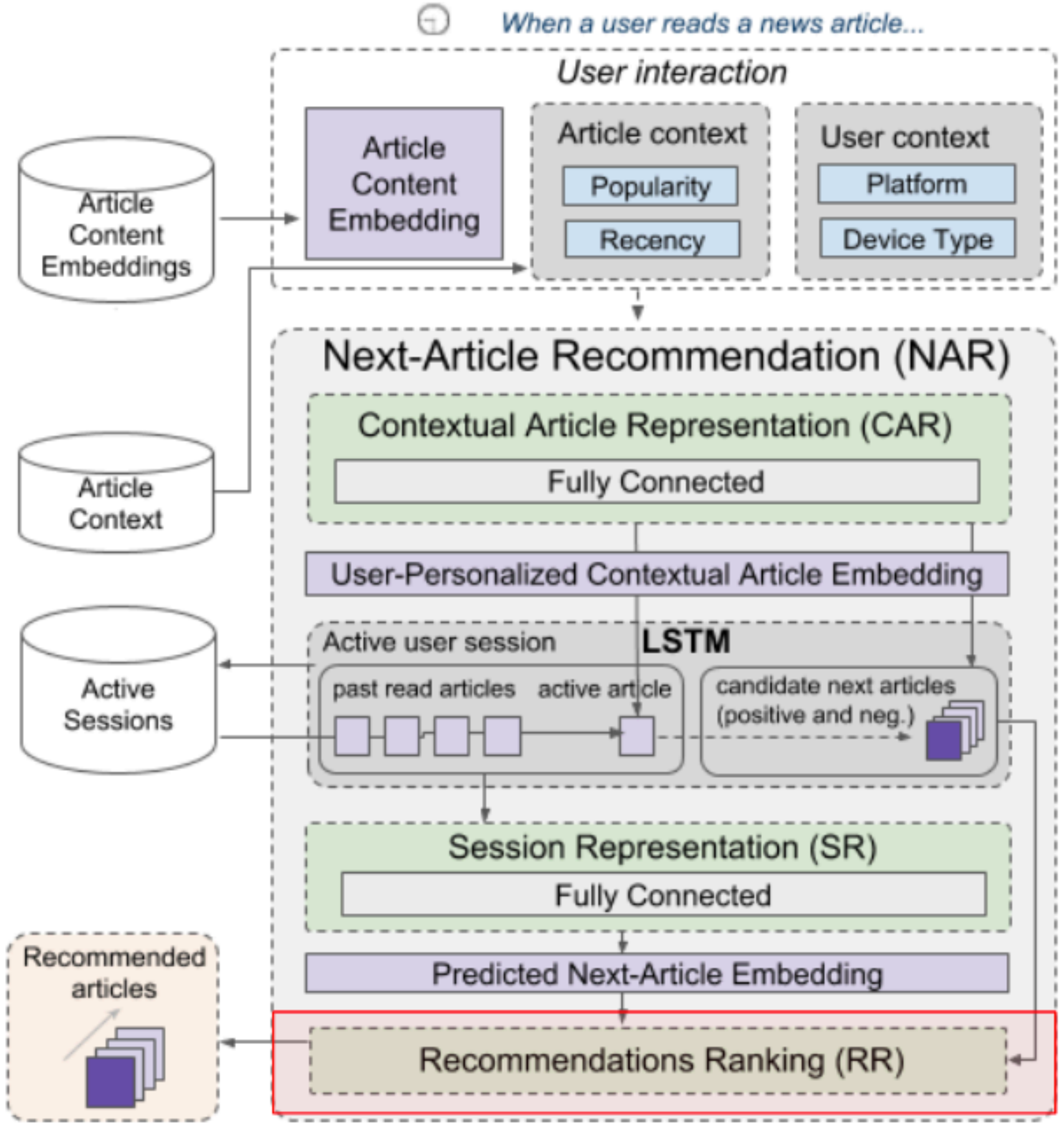
*Eq. 4 - Cosine similarity*

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

*Eq. 5 - Softmax over Relevance Score (HUANG et al., 2013)*

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

*Eq. 6 - Loss function (HUANG et al., 2013)*



05

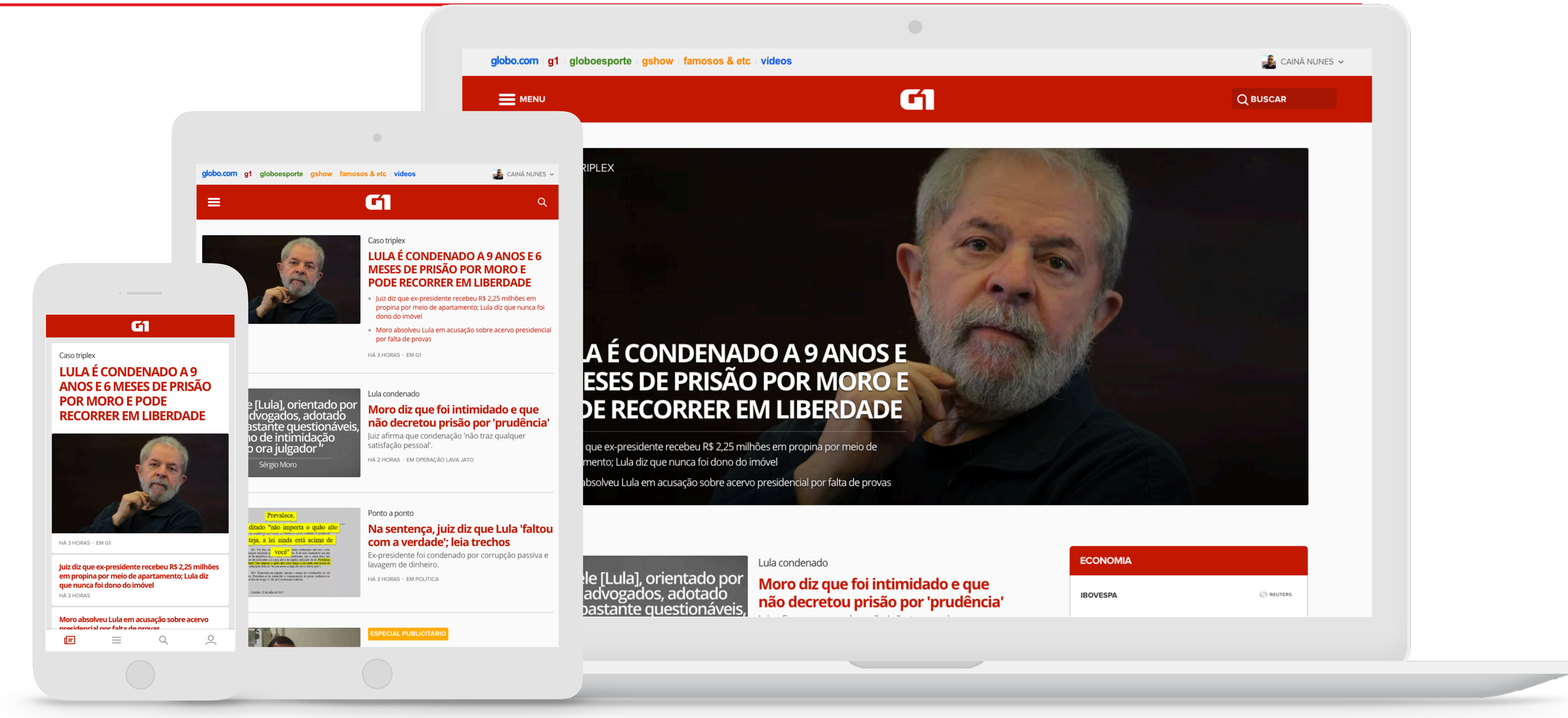
# Experiments

# Implementation

- CHAMELEON architecture instantiation implemented using TensorFlow
(available in https://github.com/gabrielspmoreira/chameleon_dlrs)

- Training and evaluation performed in Google Cloud Platform ML Engine

# Dataset

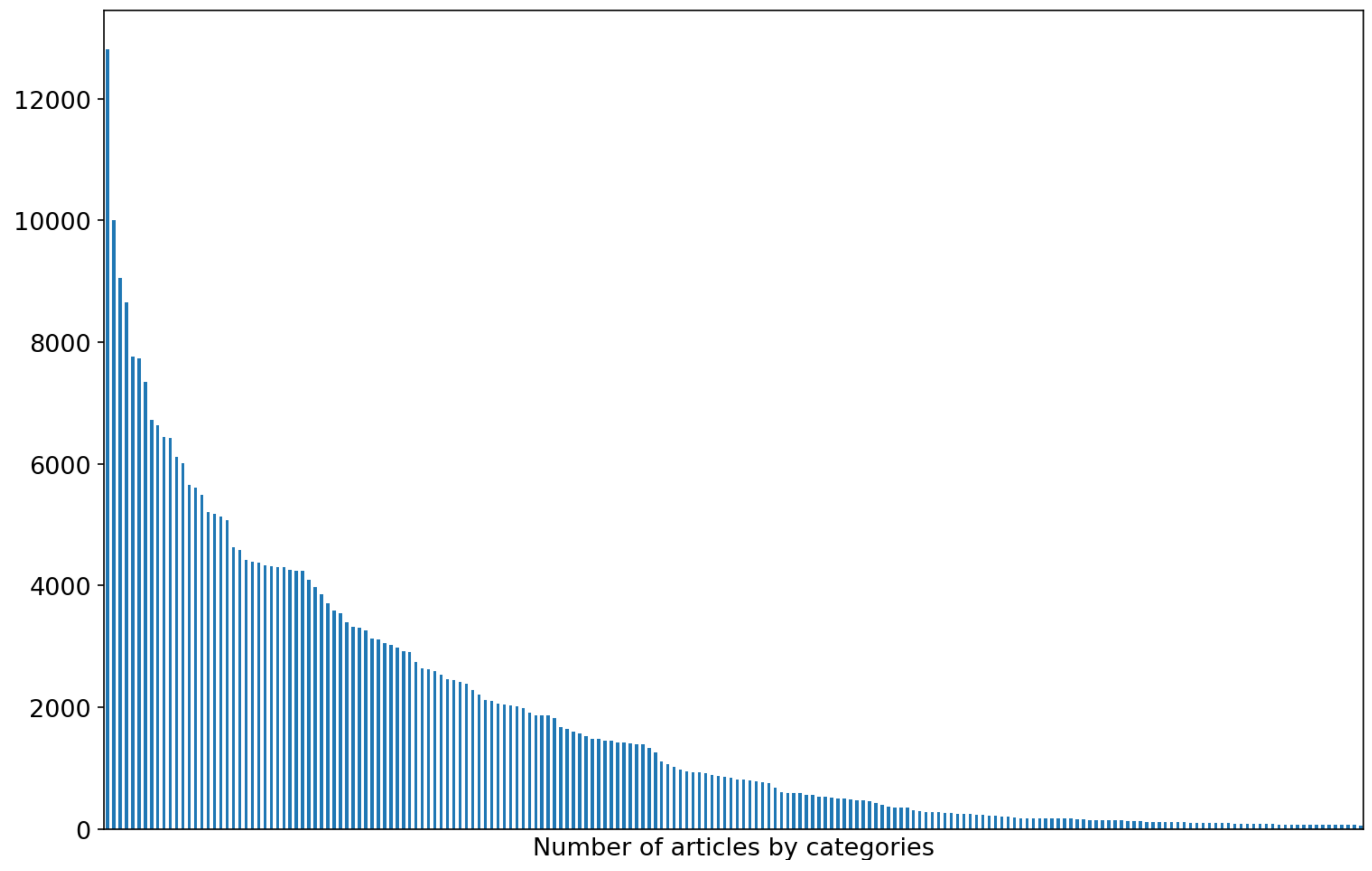



- Provided by Globo.com, the most popular news portal in Brazil

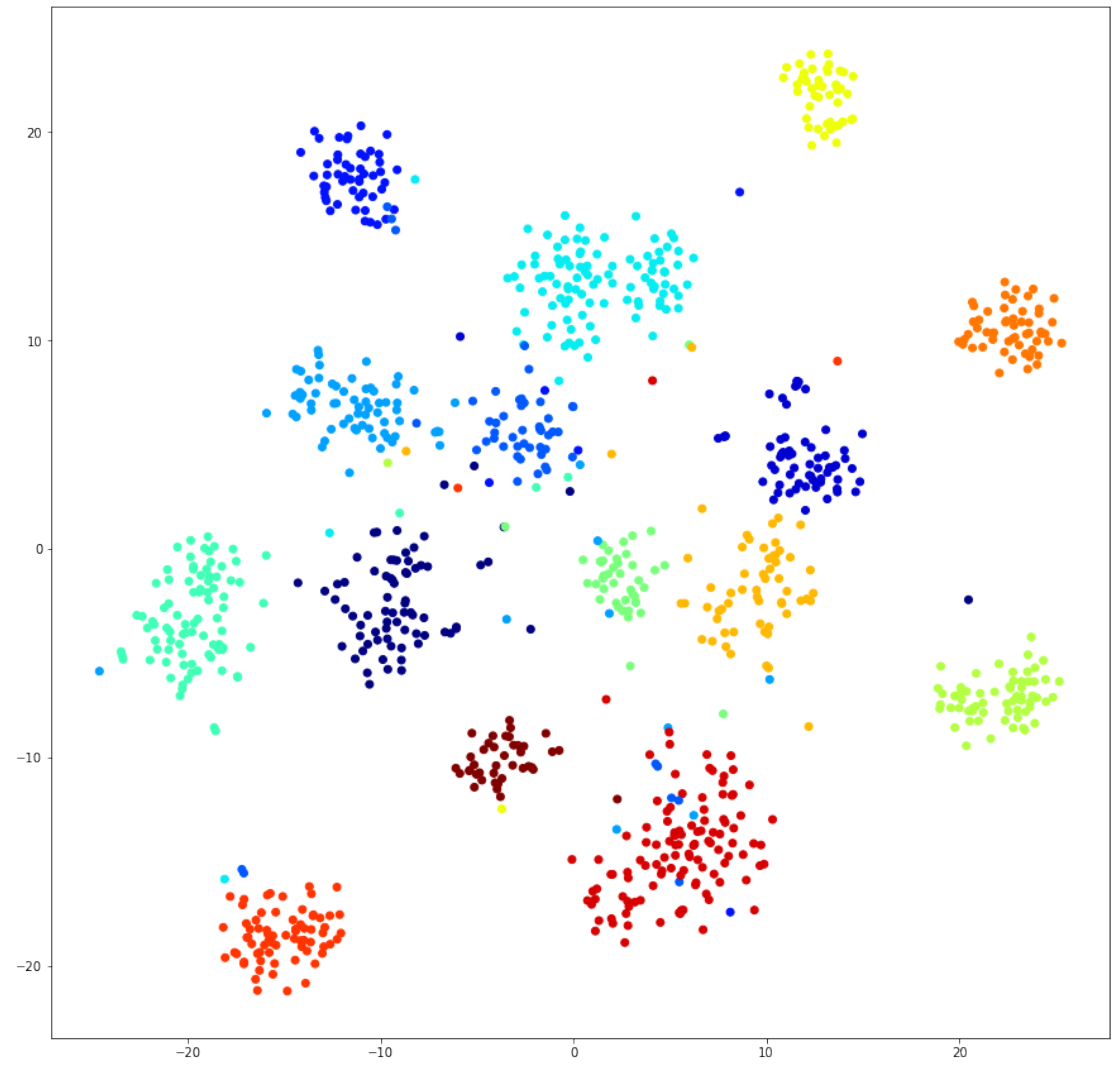
- Sample from **October, 1st - 16th , 2017**
with over 3M clicks, distributed in 1.2 M sessions from 330 K users,
who read over 50K different news articles during that period

# 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)

# NAR Module evaluation

## Temporal offline evaluation method

1. Train the NAR module with sessions within the active hour
2. Evaluate the NAR module with sessions within the next hour, for the task of the next-click prediction.

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

## Metrics

**HitRate@5 (HR@5)** - Checks whether the positive item is among the top-5 ranked items
**MRR@5 -** Ranking metric which assigns higher scores at top ranks.

# NAR Module evaluation

## Baseline methods

**1. GRU4Rec -** Seminal neural architecture using RNNs for session-based recommendations (Hidasi, 2016) with the improvements of (Hidasi, 2017) (v2).

**2. 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)

**3. 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)

# NAR Module evaluation

## Baseline 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)

# NAR Module evaluation

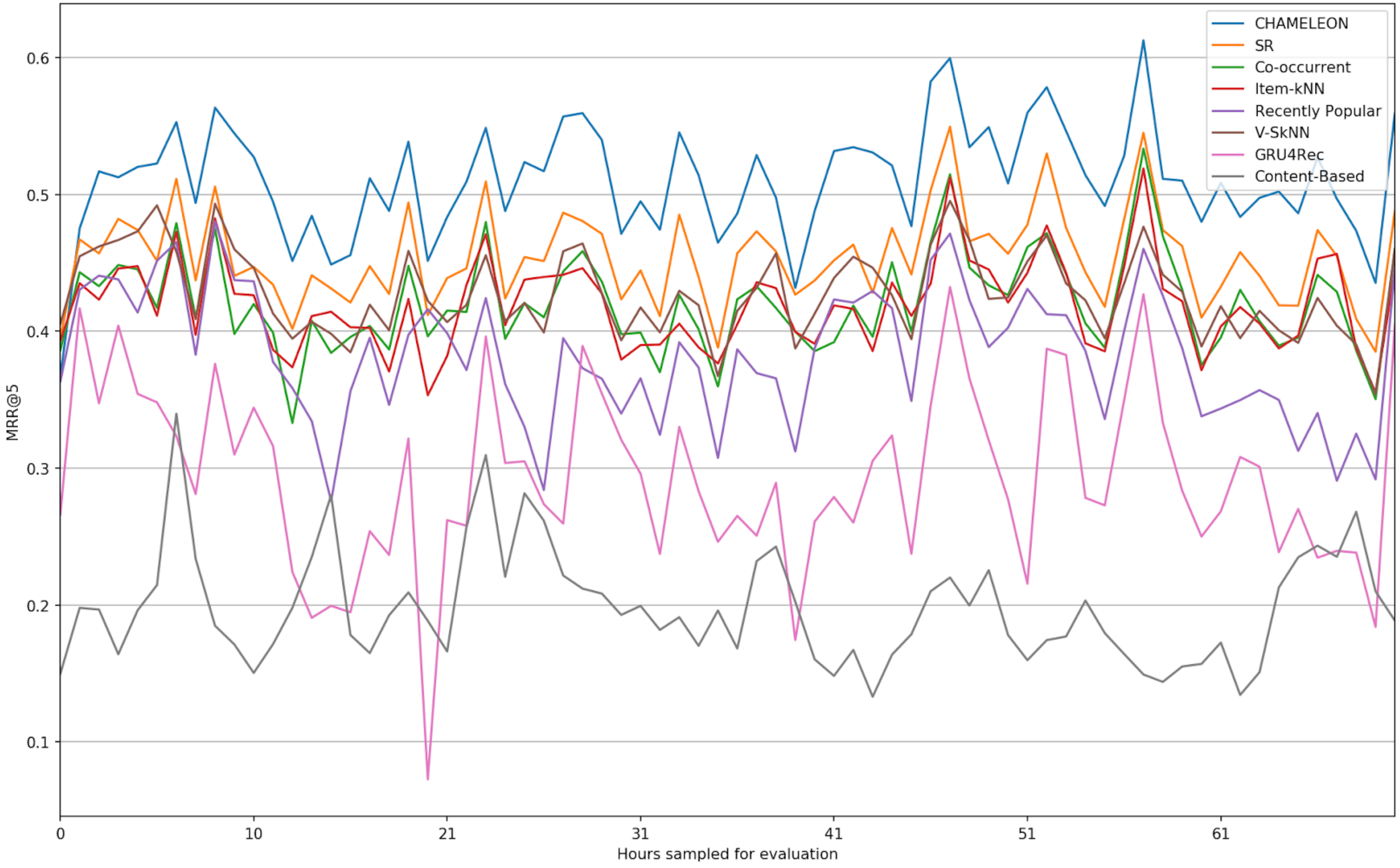
## Baseline methods

**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.

# NAR Module evaluation

Experiment #1 - Continuous training during 15 days (Oct. 1-15, 2017)

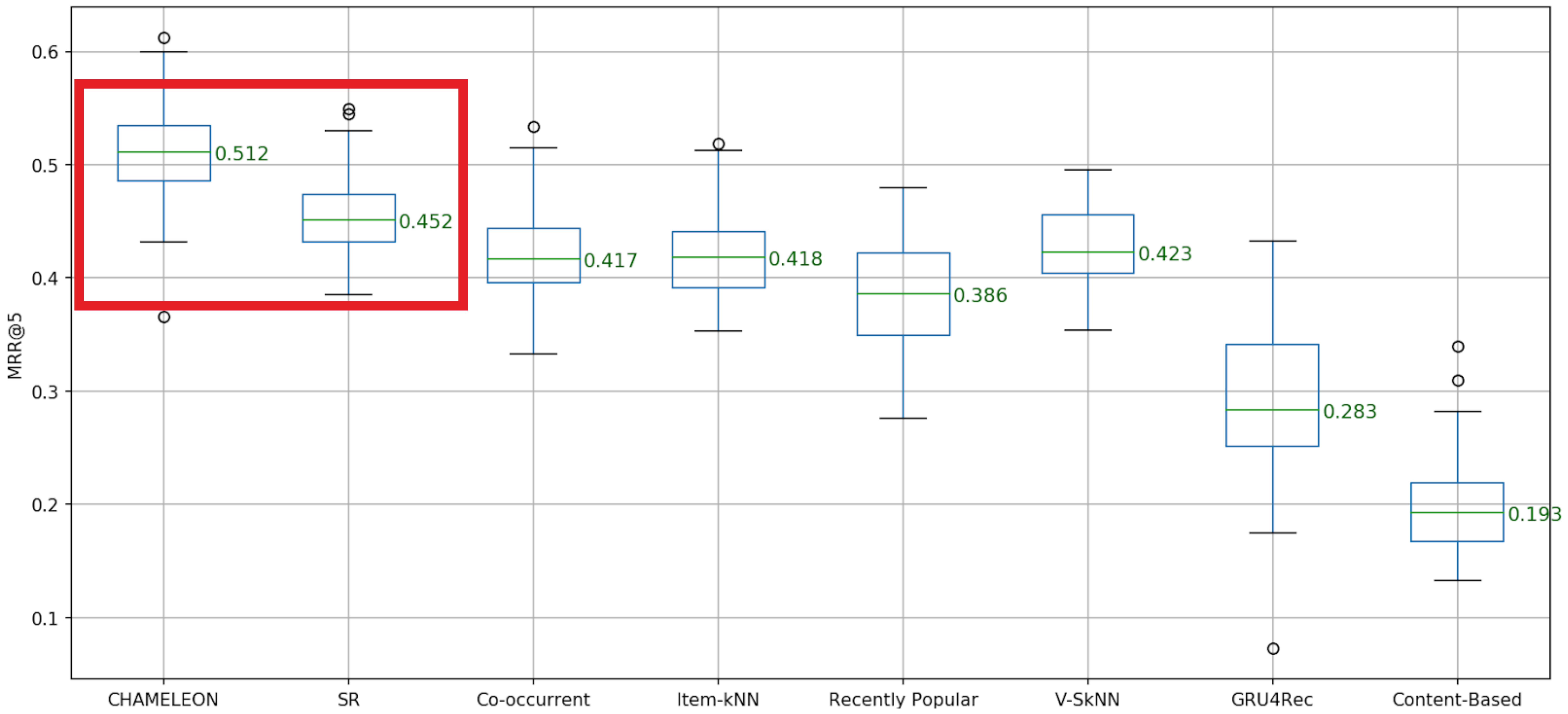



Average **MRR@5** by hour (sampled for evaluation), for a 15-days period

# NAR Module evaluation

Experiment #1 - Continuous training during 15 days (Oct. 1-15, 2017)

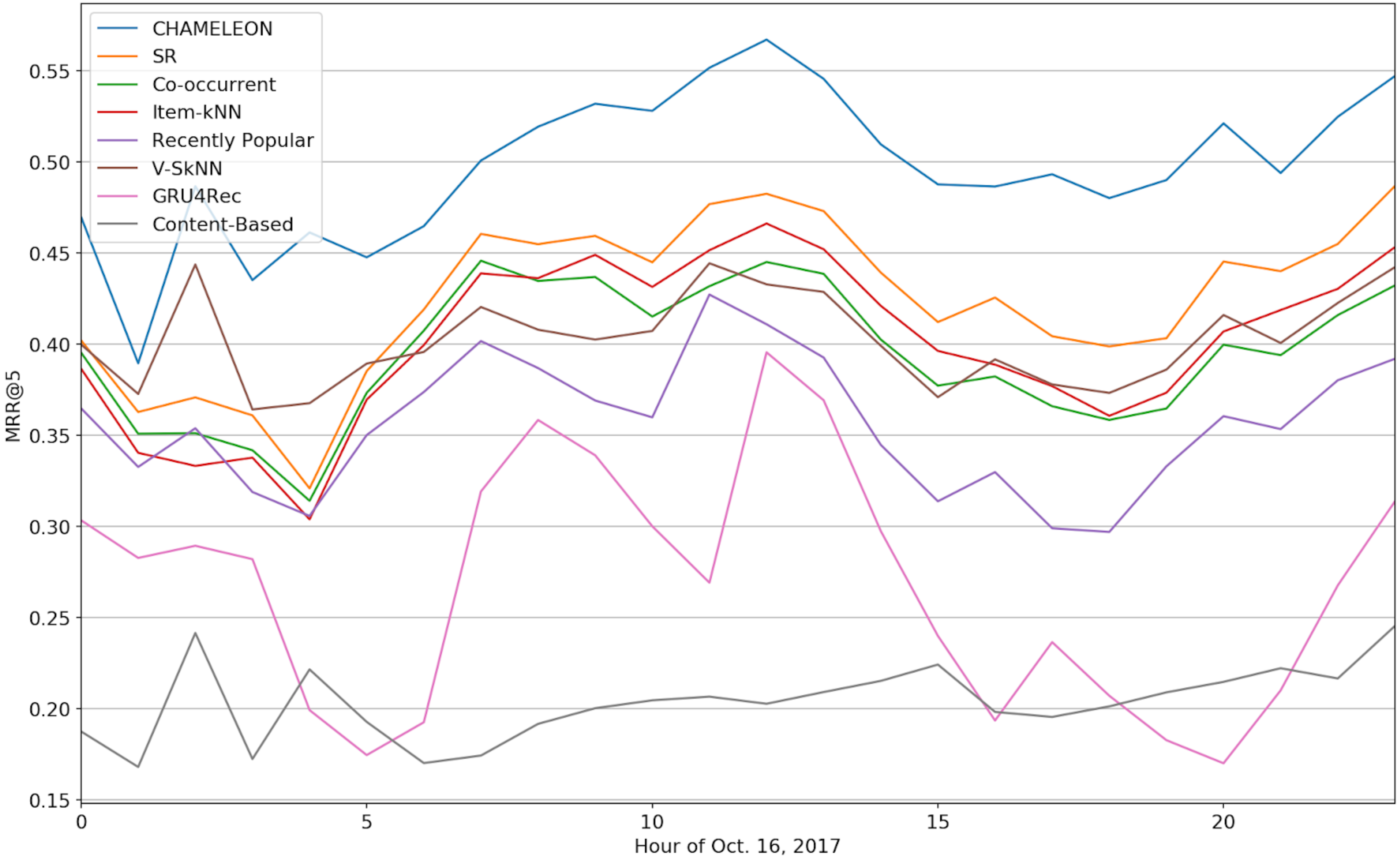
13% of relative improvement on MRR@5




Distribution of Average **MRR@5** by hour (sampled for evaluation), for a 15-days period

# NAR Module evaluation

Experiment #2 - Continuous training and evaluating each hour, on the subsequent hour (Oct. 16, 2017)




Average **MRR@5** by hour, for Oct. 16, 2017

06

# Related Work

# Main Inspirations

**GRU4Rec** (Hidasi,2016)
The seminal work on the usage of Recurrent Neural Networks (RNN) on session-based recommendations, and subsequent work (Hidasi,2017).

**MV-DNN** (Elkahky,2015)
Adapted Deep Structured Semantic Model **(DSSM)** for the recommendation task. MV-DNN maps users and items to a latent space, where the cosine similarity between users and their preferred items is maximized. That approach makes it possible to keep the neural network architecture static, rather than adding new units into the output layer for each new item.

The **MV-DNN** was also adapted for news recommendation by Temporal DSSM **(TDSSM)** (Song,2016) and Recurrent Attention DSSM **(RA-DSSM)** (Kumar,2017)

07

# Conclusion

# Conclusion

- Proposal of an instantiation of the CHAMELEON - a Deep Learning Meta-Architecture for News Recommender Systems, using a 1D CNN to extract textual features from news articles and a LSTM to model user sessions.

- Recommendations accuracy and ranking quality provided by CHAMELEON were constantly higher over time than an extensive number of baseline methods for session-based recommendation. The median HR@5 and MRR@5 obtained by CHAMELEON were 10% and 13% higher than the best baseline method.

- A temporal offline evaluation method was also proposed to emulate the dynamics of news readership, where articles context (recent popularity and recency) is constantly changing.

# Thank You


**Gabriel Moreira**

www.linkedin.com/in/gabrielspmoreira

gspmoreira@gmail.com

@gspmoreira



**Felipe Ferreira**

www.linkedin.com/in/feliferr

feliferrgo@gmail.com

@feliferrgo


code and dataset: **http://bit.ly/chameleon_dlrs**