Popularity Prediction on Online Articles with Deep Fusion of Temporal Process and Content Features

Authors: Dongliang Liao, Jin Xu, Gongfu Li,
Weijie Huang, Weiqing Liu, Jing Li

Outline

1. Introduction
2. Why is this research important?
3. Problem Formulation
4. Models Used
5. Experiments
6. Results

Introduction

- Around 2.5 quintillion bytes of data is created every day.
- Predicting what makes an online article popular is very important to social media tech companies.
- Predicting the popularity of an online article has many applications such as: recommendations, advertising concerns, and retrieval of information.

Authors of the Paper

- Research was done by employees at WeChat which is owned by Tencent.
- WeChat is a social media platform which displays over 500,000 articles per day with 2 billion views.
- Tencent is a multinational conglomerate valued at $406 billion.

Importance

- Social media’s profitability relies heavily on advertising.
- Suggesting articles with a high probability of being popular means more views, which means more advertising revenue.
- Suggesting popular articles means users will spend more time on social media, and provide the company with more data on their browsing habits.

Importance

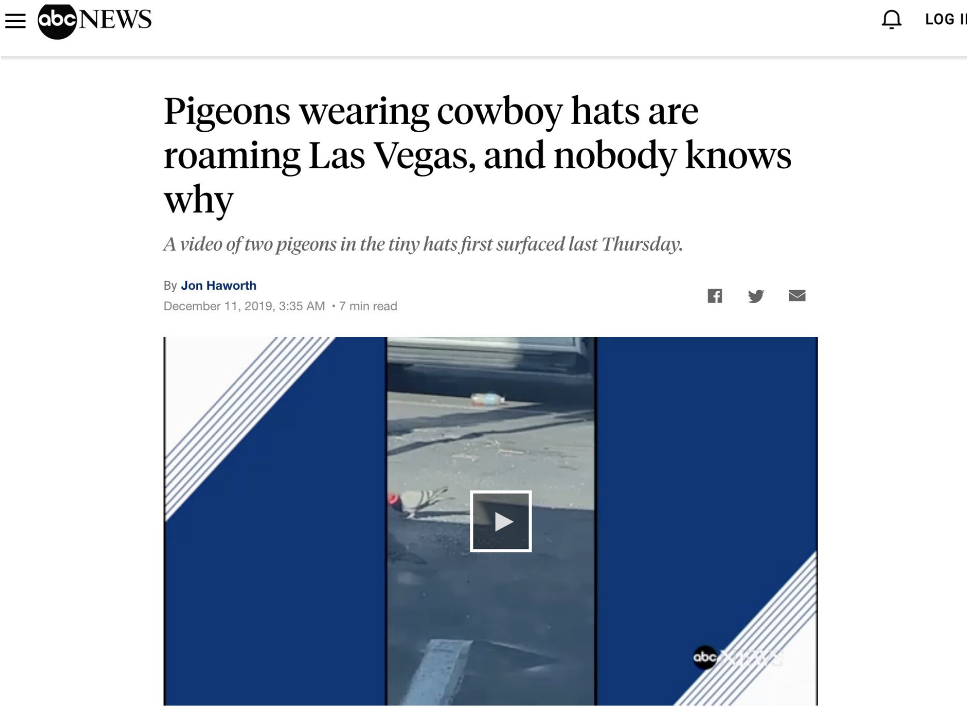
- Learning what makes an article popular can allow media companies to compose their articles in ways which make them popular.
- This gives media companies more control over the flow of information, and could be used for more nefarious purposes.

Problem Formulation

- Creating a model to predict human behavior is extremely difficult.
- Humans are complex, and there are innumerable outside influences that can impact models.
- Previous models had very poor accuracy (a previous model using linear regression was only to correctly identify 30% of articles that became popular).

Problem Formulation

- The researchers decided to make this a classification model.
- Articles were classified as “hot”, “normal”, and “cold”.
- Hot was designated as articles with more than 10,000 views, while cold was less than 100, and normal as in between the two.



Problem Formulation

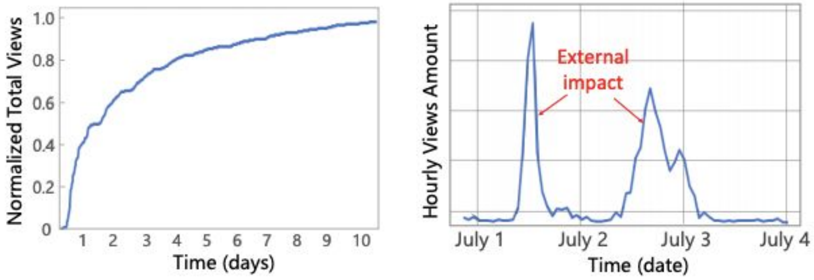
- Total views discretized into time intervals
- Continuous time also discretized into time slots
- Aggregate user feedback events which include:
    - View
    - Like
    - Comment
    - Share

Model

- The researchers employed temporal attention fusion to combine a temporal process model and content features to make a more refined model.
- Previous research had found some accuracy with a temporal process model, and a refinement of content features; so the next step was to combine the two.

Temporal Process Model

- This element of the overall model is concerned with trends and fluctuations in article views.
- It's broken down into 2 parts:
  - A Recurrent Neural Network (RNN), in this case the Long Short Term Memory (LSTM) structure.
  - A Convolutional Neural Network (CNN)



(a) Long term growth trend.

(b) Short term fluctuation.

CNN

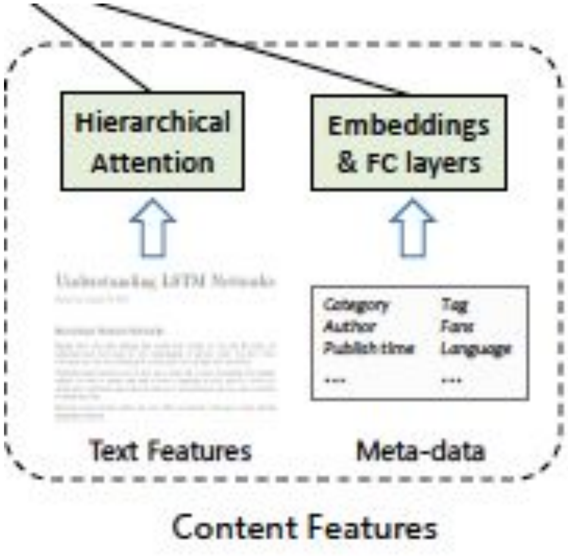
- Named Convolutional Neural Networks because of the *convolution* operation present within the neurons.
- Convolution is a matrix operation that describes how the shape of one function is modified by another in the form of a third function.
- The researchers used a CNN to model short term fluctuations in popularity.

LSTM

- Primarily designed to model temporal sequences.
- Memory states present within the model allow it to process segments of data at a time instead of individual points.
- Researchers used this model to account for growth treads on a macro scale.

Content Features

- This part of the model extracts features from the articles which will be combined with the temporal models.
- Also broken up into two parts:
    - Hierarchical Attention Network (HAN)
    - Feature combination of meta-data (FC)



HAN

- HAN was employed to extract features involving words and sentences.
- As words and sentences make up a hierarchical structure it was an obvious choice.
- These hierarchical features are encoded into a vector along with a separate vector encoded with title information.

FC

- Meta-data extracted consisted of numerical features, category, the number of fans an author might have, and if the author had previously wrote an article that was classified as hot.
- Embedding techniques are then used to place the features into vectors, and then they are embedded into fully connected layers.

Combination

- The four models are then combined into a Temporal Attention Fusion.
- A prediction is returned, and Temporal Decayed Loss is calculated (to account for the eventual loss of interest).

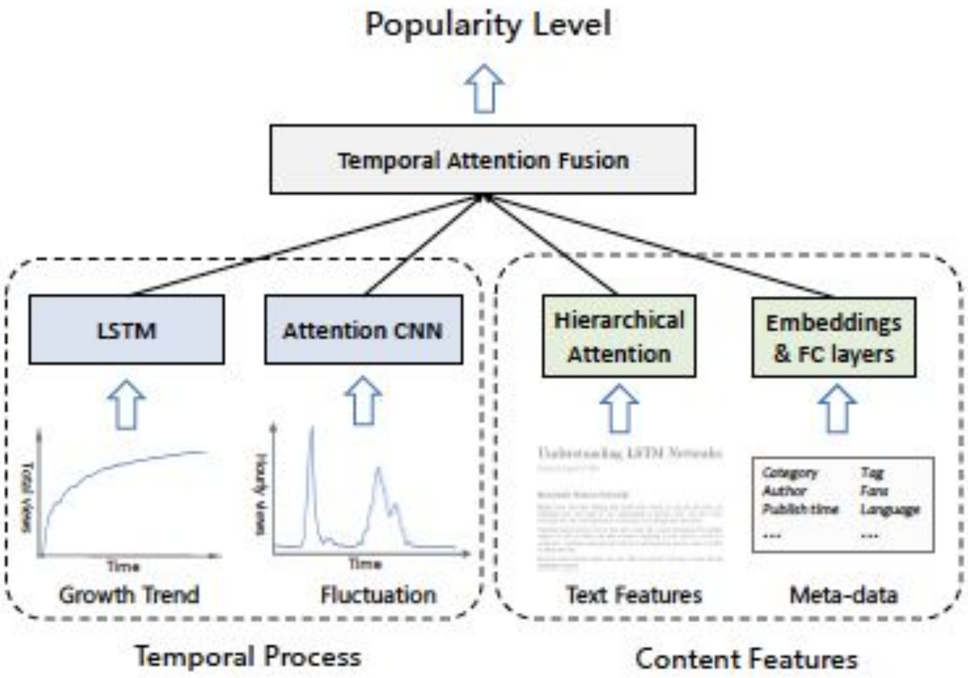
$$h_t^{merge} = \sum_{i \in \{r,c,h,e\}} \alpha_i^m \hat{h_t^i}$$

$$P_t = \mathrm{softmax}(f(h_t^{merge}))$$

$$\hat{y}_t = \arg\max_l p_t(l)$$

$$D(\Delta t) = \lceil \log_\gamma(\Delta t + 1) \rceil^{-1}$$

Combination



Experiments

- Collected all articles classified as “hot” from May 25th to July 25th.
- Collected enough “cold” and “normal” articles from the same period to create a power-law distribution (0.08% hot, 93% cold, rest normal).
- 85% of articles used to train model, 5% for validation, 10% for evaluation.
- Then tested on 30,000 articles from July 26th to August 10th.

Results

Method	Results of Balanced Test Set				Results of Random Test Set			
	Accuracy	hot F1	normal F1	cold F1	Accuracy	hot F1	normal F1	cold F1
LR	0.6441	0.3575	0.6446	0.7088	0.7551	0.4248	0.8272	0.8973
RF	0.6587	0.4246	0.6506	0.7277	0.8086	0.4743	0.8454	0.8909
HIP	0.6502	0.4353	0.6330	0.7182	0.7860	0.4342	0.7742	0.9217
VoRNN-TS	0.6709	0.4447	0.6530	0.7366	0.8569	0.4581	0.8505	0.9540
CACNN	0.6965	0.4018	0.7040	0.7394	0.8498	0.4825	0.8472	0.9493
DFTC-TS	0.7278	0.4858	0.7203	0.7638	0.8863	0.5253	0.8592	0.9698
DFTC-SF	0.6542	0.5343	0.6754	0.6212	0.6879	0.5536	0.6926	0.7869
DFTC-SM	0.7559	0.5554	0.7489	0.7812	0.9301	0.5649	0.8625	0.9759
DFTC	**0.8147**	**0.6110**	**0.7822**	**0.8393**	**0.9653**	**0.6292**	**0.8729**	**0.9916**

Table 2: Overall Prediction Performance

Questions?