Stock Movement Prediction from Tweets and Historical Prices

ACL 2018 - Yumo Xu, Shay B. Cohen

Alumno: Juan Antonio López Rivera

Dificultades del problema

El mercado de acciones es:

- Complejo
- Estocástico
- Caótico
- Dependencias temporales

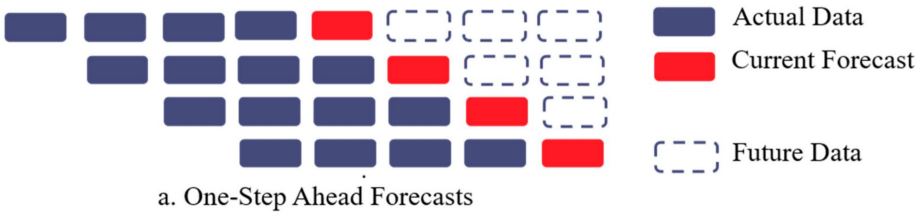
StockNet

Objetivo

Predecir el movimiento de cierta acción en el día $\mathcal{D}$

Usando datos históricos de Twitter y precio de la acción entre los días

$$[\mathcal{D} - \Delta\mathcal{D}, \mathcal{D} - 1]$$



a. One-Step Ahead Forecasts

StockNet

Datos de entrada

Entre las fechas 01/01/2014 a 01/01/2016 se recolectaron datos de 88 compañías.

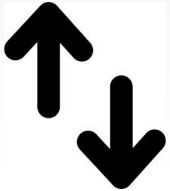
Para cada compañía se tiene:

- Tweets que mencionan esa acción
- Precio histórico

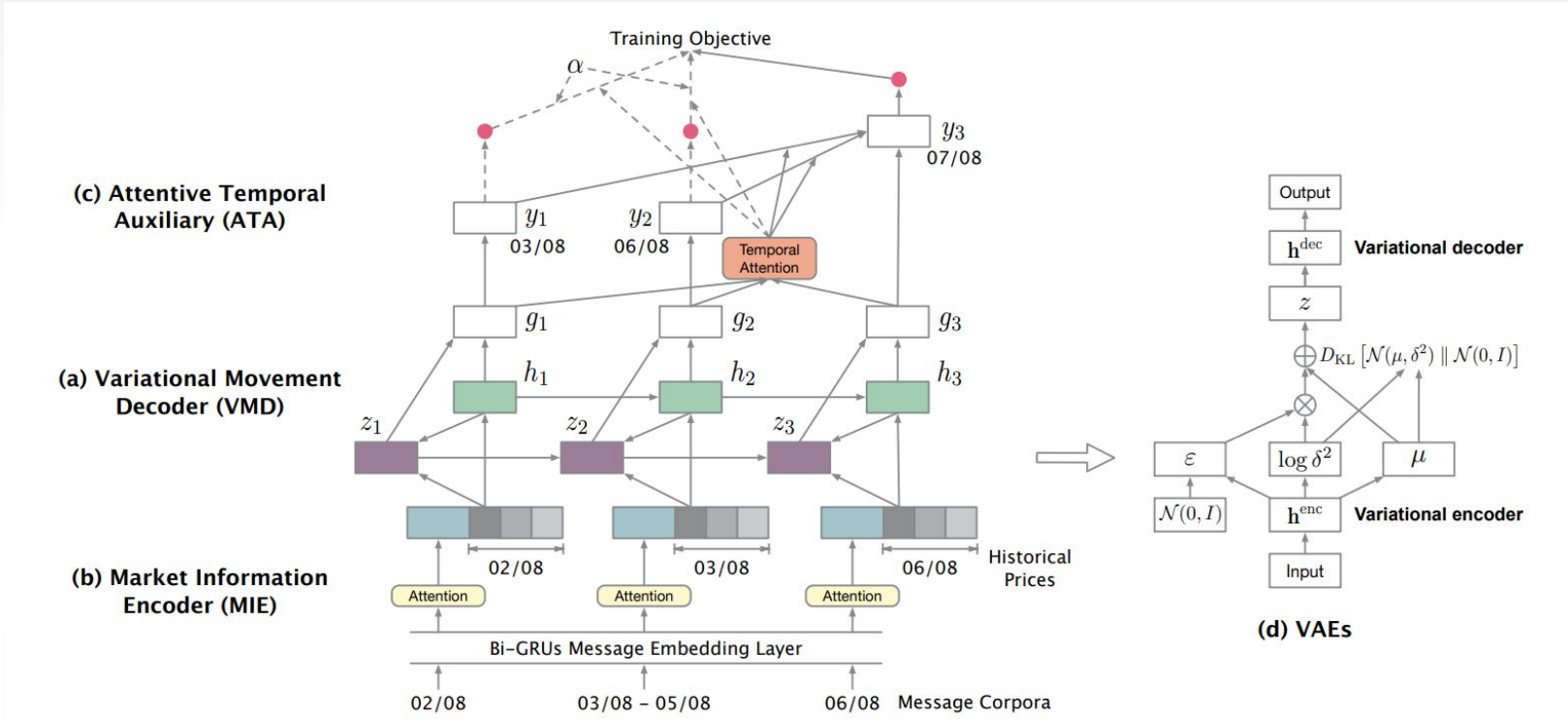
Twitter API

Salida

- 0: Bajará la acción
- 1: Subirá la acción



StockNet


Training Objective
α
y3
07/08
y1
03/08
y2
06/08
Temporal Attention
(c) Attentive Temporal Auxiliary (ATA)
g1
g2
g3
h1
h2
h3
(a) Variational Movement Decoder (VMD)
z1
z2
z3
02/08
03/08
06/08
Historical Prices
Attention
Attention
Attention
(b) Market Information Encoder (MIE)
Bi-GRUs Message Embedding Layer
02/08
03/08 – 05/08
06/08
Message Corpora
Output
h^dec
Variational decoder
z
D_KL [N(μ, δ²) ‖ N(0, I)]
ε
log δ²
μ
N(0, I)
h^enc
Variational encoder
Input
(d) VAEs

0

Recolección de datos

Historical Price Dataset

Twitter Dataset

Recolección de datos (1): Precio histórico

- 88 compañías de alto volumen de movimientos
  - Top 10 con mayor capital en cada categoría
- El movimiento debe ser significativo, ie. ≤-0.5% o bien >0.55%
  - Descarta 38.72% de los datos, quedando 26,614
  - Balancea las clases: 49.78% y 50.22%

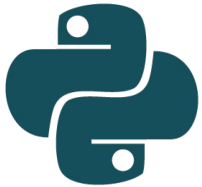
Recolección de datos (1): Precio histórico

- **Entrenamiento**: 20,399 movimientos entre 01/01/2014 y 01/08/2015
- **Validación**: 2,555 movimientos entre 01/08/2015 y 01/10/2015
- **Prueba**: 3,720 movimientos entre 01/10/2015 y 01/01/2016

Recolección de datos (2): Twitter Dataset

- Filtrar con búsquedas REGEX en Twitter API todos los tweets en el periodo 01/01/2014 a 01/01/2016 que mencionan el símbolo de NASDAQ correspondiente
  - Por ejemplo: "$GOOG" (Google), "$AAPL" (Apple), etc.
- Preprocesar tweets con el paquete NLTK (Natural Language Toolkit)
  - Tokenización, tratamiento de hipervínculos, "@", hashtags

Twitter API



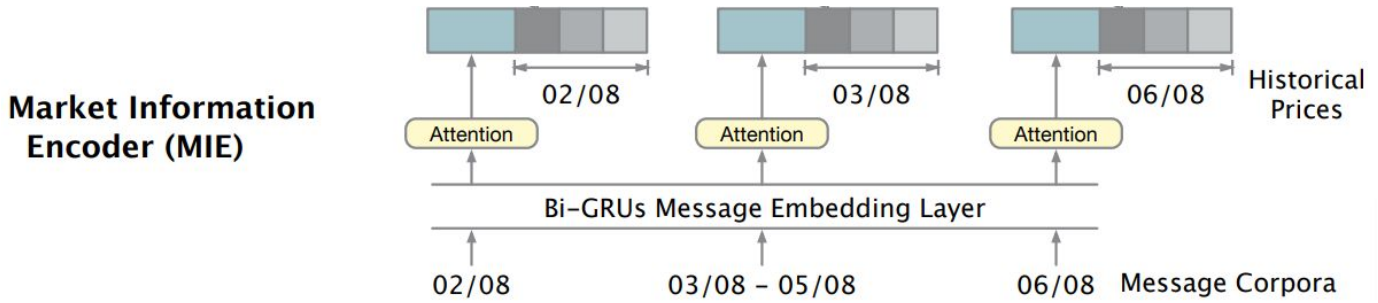
NLTK

1

Codificación de datos

Market Information Encoder (MIE)

- Codifica información de redes sociales y precio histórico para formar la **entrada X**
- Cada día corresponde a una entrada **xt = [ct , pt]** donde
  - **ct**: embedding del corpus
  - **pt**: vector de precio histórico



Market Information Encoder (MIE)

- El vector de precio histórico pt se conforma de 3 valores:
  - Precio al cierre del día
  - Precio más alto del día
  - Precio más bajo del día

$$\tilde{p}_t = \left[\tilde{p}_t^c, \tilde{p}_t^h, \tilde{p}_t^l\right]$$

- Finalmente se normalizan usando el cierre del día anterior:

$$p_t = \tilde{p}_t / \tilde{p}_{t-1}^c - 1$$

Market Information Encoder (MIE)

Message Embedding Layer

- Para cada tweet del corpus

  ℓ^* es la posición del símbolo del stock correspondiente ($...)
- Se ejecutan 2 GRUs en direcciones opuestas a partir de ℓ^*
- Se promedia el estado final de ambas
- Todos los m de un día conforman la Message Embedding Matrix:

$$M_t \in \mathbb{R}^{d_m \times K}$$

$$\overrightarrow{h}_f = \overrightarrow{\text{GRU}}(e_f, \overrightarrow{h}_{f-1})$$
$$\overleftarrow{h}_b = \overleftarrow{\text{GRU}}(e_b, \overleftarrow{h}_{b+1})$$
$$m = (\overrightarrow{h}_{\ell^\star} + \overleftarrow{h}_{\ell^\star})/2$$

where $f \in [1, \ldots, \ell^\star]$, $b \in [\ell^\star, \ldots, L]$

Market Information Encoder (MIE)

Message Embedding Layer (2)

- La calidad y relevancia de los tweets varía drásticamente
- Por ello se pondera la matriz **Mt** calculando el vector u_t:

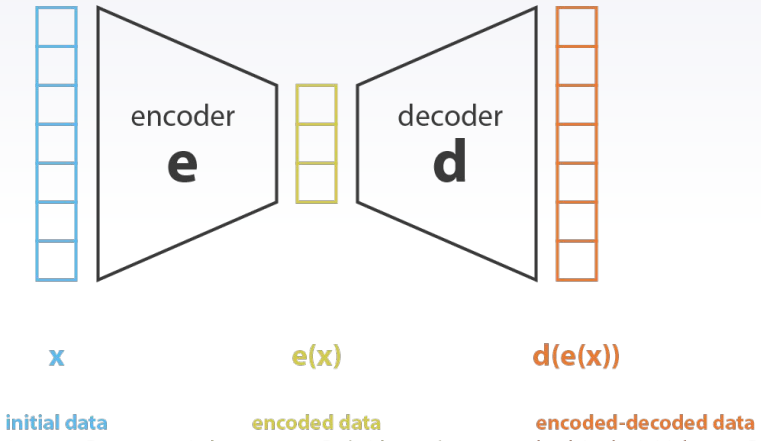
$$u_t = \zeta(w_u^\intercal \tanh(W_{m,u} M_t))$$

- Finalmente se multiplican ambos para obtener el embedding de corpus:

$$c_t = M_t u_t^\intercal.$$

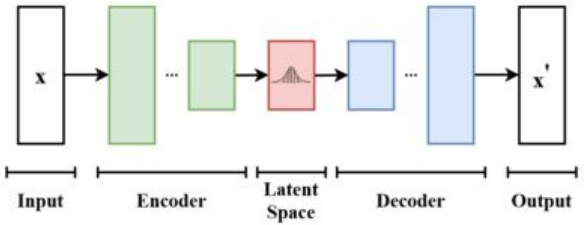
2

Variational Movement Decoder (VMD)

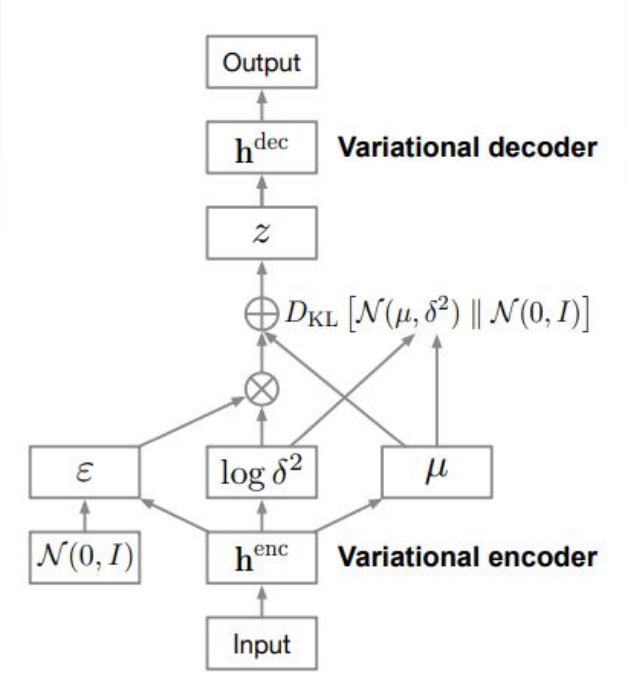


Variational Autoencoders

simple autoencoders	input **x**	encoding →	latent representation **z = e(x)**	decoding →	input reconstruction **d(z)**	
variational autoencoders	input **x**	encoding →	latent distribution **p(z|x)**	sampling →	sampled latent representation **z ~ p(z|x)**	decoding → input reconstruction **d(z)**



Variational Autoencoders

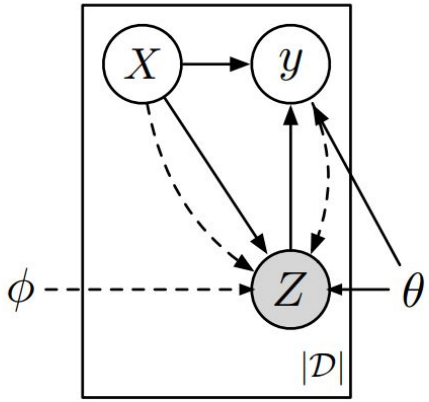


(d) VAEs

DKL: Divergencia Kullback-Leibler entre dos distribuciones

Variational Movement Decoder (VMD)

- Infiere **Z** dado **X, y**
- Decodifica el movimiento **y** a partir de **X, Z**



$$p_\theta(y|X) = \int_Z p_\theta(y, Z|X)$$

$$p_\theta(y, Z|X) = p_\theta(y_T|X, Z)\, p_\theta(z_T|z_{<T}, X) \quad (2)$$

$$\prod_{t=1}^{T-1} p_\theta(y_t|x_{\leq t}, z_t)\, p_\theta(z_t|z_{<t}, x_{\leq t}, y_t)$$

- **Líneas sólidas**: proceso generativo
- **Líneas punteadas**: aproximación variacional al *posterior intractable*

Variational Movement Decoder (VMD)

- El valor **Z** se conoce como ***latent driven factor***, se aproxima así (cuando no se conoce ***y***) en cada instante de tiempo *zt*:

$$z_t = \mu_t + \delta_t \odot \epsilon$$

$$\mu'_t = W^{\theta}_{o,\mu} h^{z\prime}_t + b^{\theta}_{\mu}$$

$$\log \delta'^2_t = W^{\theta}_{o,\delta} h^{z\prime}_t + b^{\theta}_{\delta}$$

Variational Movement Decoder (VMD)

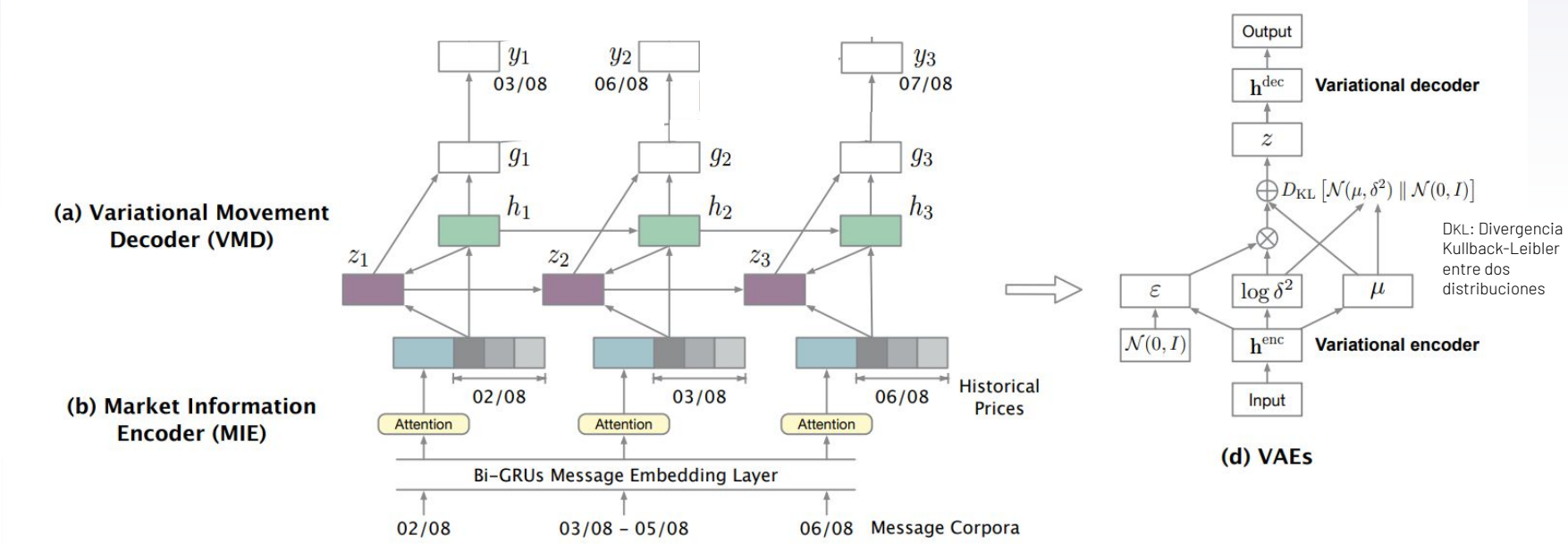
- VMD incorpora una celda GRU para extraer un estado *h* de características en cada instante de tiempo:

$$h_t^s = \mathrm{GRU}(x_t, h_{t-1}^s)$$

- La hipótesis final en cada instante es dada por:

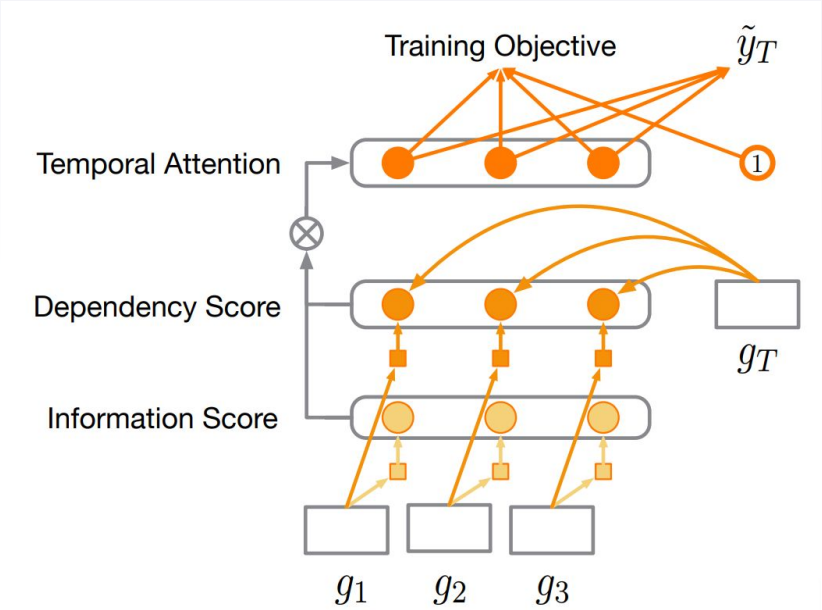
$$g_t = \tanh(W_g[x_t, h_t^s, z_t] + b_g)$$
$$\tilde{y}_t = \zeta(W_y g_t + b_y), t < T$$
$$\tilde{Y}^* = [\tilde{y}_1; \ldots; \tilde{y}_{T-1}]$$

Variational Movement Decoder (VMD)



3

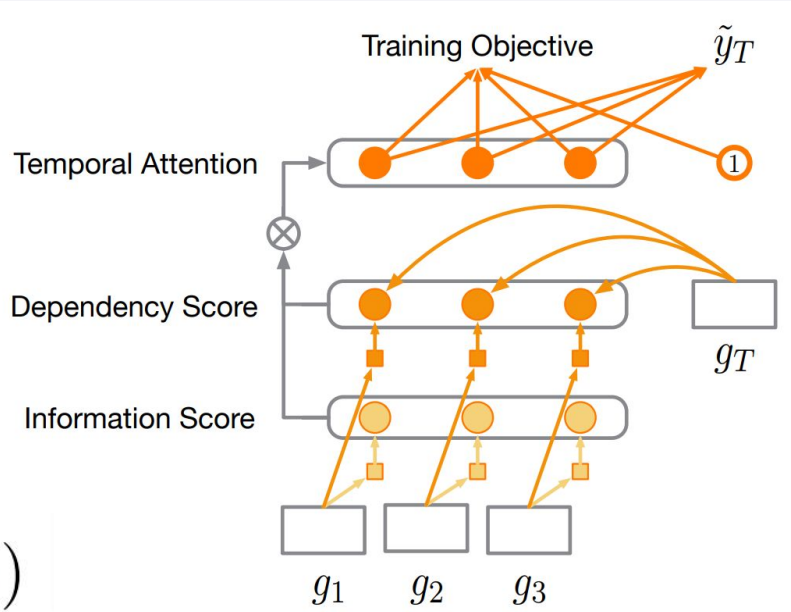
Attentive Temporal Auxiliary (ATA)



Attentive Temporal Auxiliary (ATA)

$$v_i' = w_i^\intercal \tanh(W_{g,i} G^*)$$
$$v_d' = g_T^\intercal \tanh(W_{g,d} G^*)$$
$$v^* = \zeta(v_i' \odot v_d')$$

Normalized Attention Weight



$$\tilde{y}_T = \zeta(W_T[\tilde{Y}^* v^{*\intercal}, g_T] + b_T)$$

Main Hypothesis

$$\tilde{Y}^* = [\tilde{y}_1; \ldots; \tilde{y}_{T-1}]$$

Un último detalle

To incorporate varied temporal importance at the objective level, we first break down the approximated $\mathcal{L}$ into a series of temporal objectives $f \in \mathbb{R}^{T\times 1}$ where f_t comprises a likelihood term and a KL term for a trading day t,

$$f_t = \log p_\theta\left(y_t | x_{\leq t}, z_{\leq t}\right) - \lambda D_{\mathrm{KL}}\left[q_\phi\left(z_t | z_{<t}, x_{\leq t}, y_t\right) \parallel p_\theta\left(z_t | z_{<t}, x_{\leq t}\right)\right] \quad (27)$$

$$\alpha \in [0, 1]$$

$$v = [\alpha v^*, 1]$$

Final temporal weight vector

$$\mathcal{F}(\theta, \phi; X, y) = \frac{1}{N}\sum_{n}^{N} v^{(n)} f^{(n)}$$

Training objective

$$\tilde{y}_T = \zeta(W_T[\tilde{Y}^* v^{*\intercal}, g_T] + b_T)$$

Main Hypothesis

$$\tilde{Y}^* = [\tilde{y}_1; \ldots; \tilde{y}_{T-1}]$$

4 Resultados

Parámetros

- Trading days: 5
- Batch Size: 32
- Max. tweets per day: 40
- Max. tokens per tweet: 30
- Word embedding size: 50
- Message Embedding Layer: 100
- Variational Movement Decoder: 150
- Adam optimizer
- Learning rate: 0.001
- Input dropout rate: 0.3

"to control memory costs and make model training feasible on one single GPU (11GB memory)"

Métricas

$$\textbf{\textit{Accuracy}} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\mathrm{MCC} = \frac{\mathrm{tp} \times \mathrm{tn} - \mathrm{fp} \times \mathrm{fn}}{\sqrt{(\mathrm{tp} + \mathrm{fp})(\mathrm{tp} + \mathrm{fn})(\mathrm{tn} + \mathrm{fp})(\mathrm{tn} + \mathrm{fn})}}$$

Resultados

Baseline models	Acc.	MCC	StockNet variations	Acc.	MCC
RAND	50.89	-0.002266	TECHNICALANALYST	54.96	0.016456
ARIMA (Brown, 2004)	51.39	-0.020588	FUNDAMENTALANALYST	58.23	0.071704
RANDFOREST (Pagolu et al., 2016)	53.08	0.012929	INDEPENDENTANALYST	57.54	0.036610
TSLDA (Nguyen and Shirai, 2015)	54.07	**0.065382**	DISCRIMINATIVEANALYST	56.15	0.056493
HAN (Hu et al., 2018)	**57.64**	0.051800	HEDGEFUNDANALYST	**58.23**	**0.080796**

- **RAND**: Random guesses
- **ARIMA**: Analysis with only historical prices
- **RANDFOREST**: Random forest with word2vec
- **TSLDA**: Generative model that learns topics and sentiments
- **HAN**: State of the art DNN with hierarchical attention

- **Technical Analyst**: only historical prices
- **Fundamental Analyst**: only tweet information
- **Independent Analyst**: without ATA
- **Discriminative Analyst**: without Z (zt = μt)
- **Hedge Fund Analyst: full STOCKNET**

Efecto del Attention Temporal Auxiliary

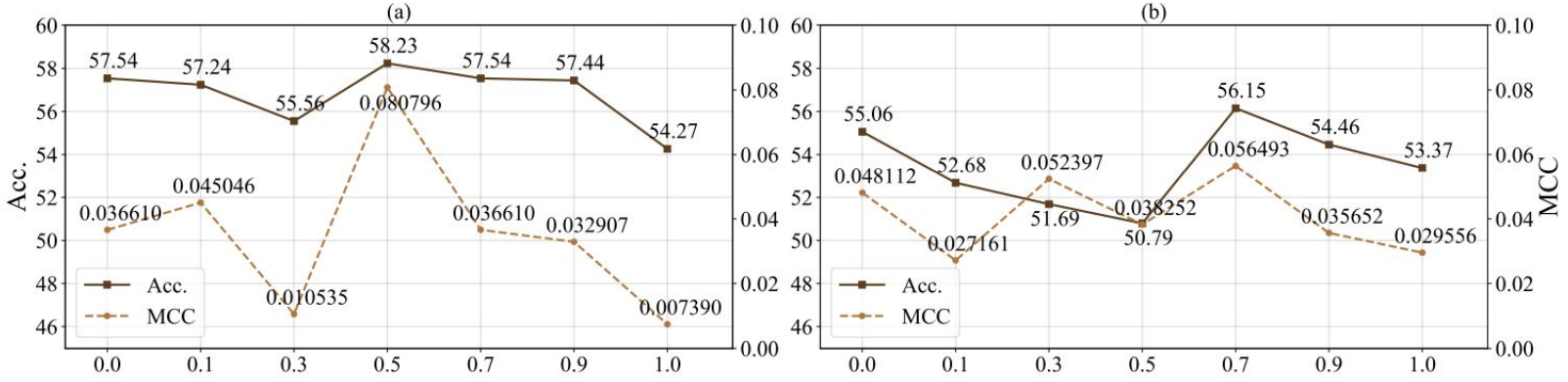
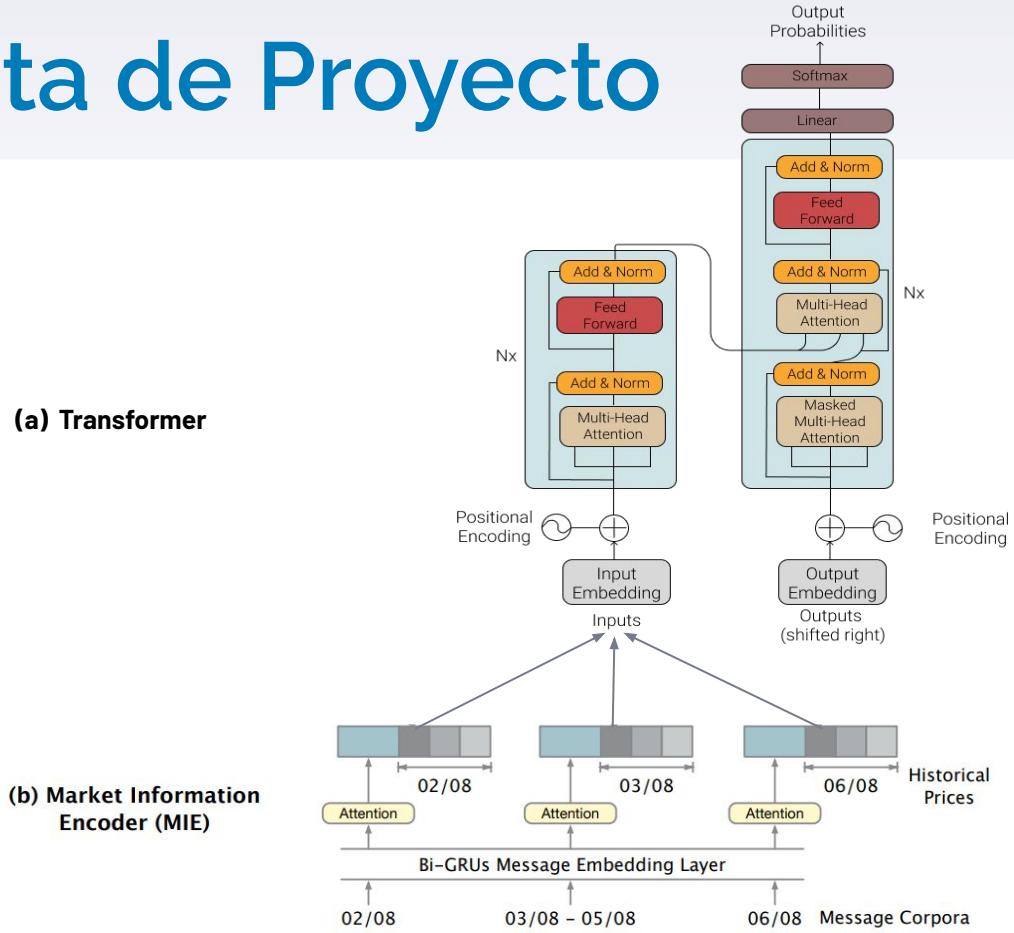


Figure 4: (a) Performance of HEDGEFUNDANALYST with varied α, see Eq. (28). (b) Performance of DISCRIMINATIVEANALYST with varied α.

$$v = [\alpha v^*, 1]$$
$$\alpha \in [0, 1]$$

5 Conclusión

Propuesta de Proyecto



¡Gracias por su atención!