# Sentiment Analysis for Official Statistics

**Francesco Pugliese, PhD**

***Italian National Institute of Statistics, Division "Information and Application Architecture", Directorate for methodology and statistical design***

Email **Francesco Pugliese** : francesco.pugliese@istat.it

# Introduction: Motivation and Goals

- Nowadays more and more people are using Social Media platforms to find out news, to express their feelings and to share or debate opinions about virtually every possible topic
  - The interest towards Social Media as a means for "measuring" public's mood is still growing
- We are investigating whether social media messages may be successfully exploited to develop *domain-specific* sentiment indices. The aim is to assess the Italian mood about *specific* topics or aspects of life, e.g.
  - the economic situation, the European Union, the migrants' phenomenon, the terrorist threat, and so on
- These new indices would enable high-frequency (e.g. daily) measures of the Italian sentiment about phenomena which are of interest in Official Statistics
- The hope is that such indices could either improve the performance of Istat's forecasting models, or enrich existing statistical products (e.g. the BES), or even be disseminated as new statistical outputs in their own right

**Author: Diego Zardetto**

# The Quest for Relevance: Filters (1/2)

- We developed procedures to collect and process only social media messages containing at least one keyword belonging to a specific 'filter', namely a definite set of relevant Italian words
  - Ideally, filters should be able to capture relevant messages and eliminate off-topic ones since the beginning
  - Domain-specific filters have been designed by subject-matter experts (possibly supported by data-driven techniques)
- At the moment we are using **Twitter** as a source, but further Social Media might be taken into consideration in the future
- Two filters have been up and running since late February 2016:

1) **'Social Mood on Economy' filter.** Designed to measure the Italian sentiment on the state of the economy. It collects ~40'000 tweets/day
2) **'Istat' filter.** Designed to enable custom downstream analyses via further filtering, and for diagnostic/validation purposes. It collects ~170'000 tweets/day

**Author: Diego Zardetto**

# The Quest for Relevance: Filters (2/2)

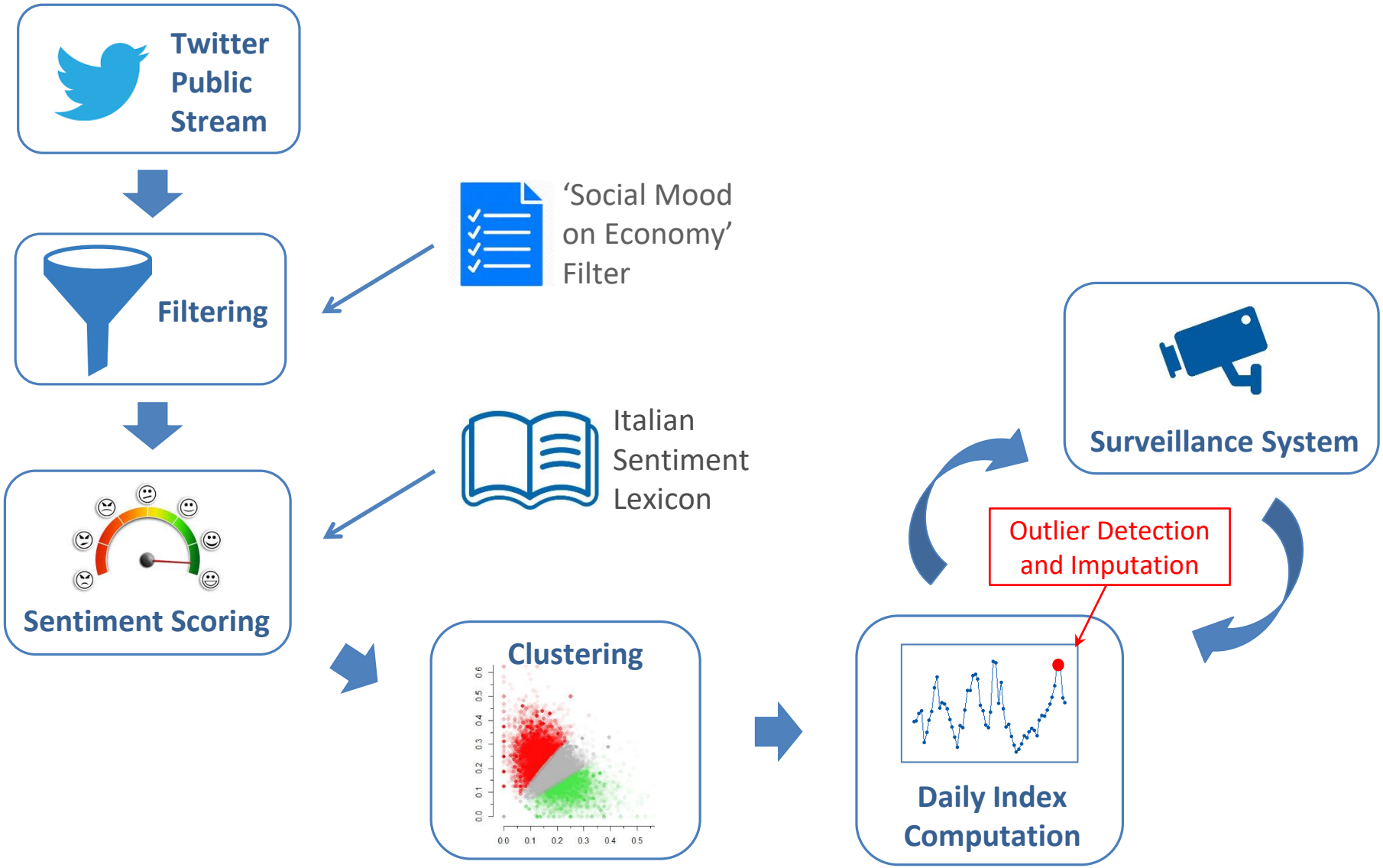
- The **'Social Mood on Economy' filter** encompasses 60 keywords (actual words or phrases). Most of these keywords have been borrowed from questionnaire items of the Italian Consumer Confidence Survey
  - However, the phenomenon tracked by the Social Mood on Economy index and consumer confidence **only partially overlap**
    - ➡ Still, the new index can detect events that influence consumer confidence but are missed by the official survey, e.g. the Central Italy earthquake of 24th Aug. 2016

- The **'Istat' filter** involves 278 keywords. These have been derived from the Themes which can be used to browse Istat's online data warehouse (I.stat)
  - Messages sampled through the 'Istat' filter are meant to represent a small-scale model of the overall population of messages which are potentially relevant in the Official Statistics perspective
    - ➡ Allows to assess the performance of the 'Social Mood on Economy' filter by providing access to "unseen" messages, i.e. those that have been filtered out

- *The rest of this presentation will focus on the Social Mood on Economy index*

**Author: Diego Zardetto**

# Processing Pipeline at a Glance




**Author: Diego Zardetto**

# Data Collection and Storage

- **Data Collection Technique**
  - We exploit Twitter's Streaming API to get low latency access to Twitter's firehose and collect samples of public tweets
- **Target Population**
  - Public tweets whose text matches at least one keyword belonging to the filter
- **Sampling Design**
  - The sampling algorithm is a black box, as it is entirely controlled by Twitter's Streaming API. At most a 1% of all the tweets produced on Twitter at a given time can be sampled
- **Data Format**
  - Twitter's Streaming API returns data in JSON format
- **Data Staging and Storage**
  - We temporarily store gathered JSON data as text files inside a staging area residing on a server. Then we periodically load bunches of tweets into an Oracle DB (and remove the corresponding files from the staging area)

**Author: Diego Zardetto**

# Text Processing and Sentiment Analysis

- **Process Granularity**
  - To compute daily index values, we process all the tweets collected in a single day as a single block
- **Input Data**
  - We only analyze the textual content of the tweets. (No information about users is ever accessed: the index only uses *unlinked anonymized* data)
- **Text Cleaning and Normalization**
  - We perform standard NLP pre-processing steps: (i) convert to lowercase, (ii) tokenize running text into words, (iii) apply basic orthographic repairs, (iv) remove URLs, (v) remove non-alphabetic characters (e.g. '#' or '@'), (vi) remove stop words, (vi) *if needed*, stem words to get rid of inflected forms
- **Sentiment Analysis Approach**
  - To classify tweets as Positive, Negative or Neutral we chose to adopt an unsupervised, lexicon-based approach
  - We discarded supervised, Machine Learning approaches because we were unable to find large, high quality training sets of human-labeled tweets in Italian

**Author: Diego Zardetto**

# Sentiment Scores: The Lexicon

- Our Sentiment Analysis application involves two sequential steps
  1) Calculate sentiment scores for each tweet
  2) Use these sentiment scores to cluster tweets into three mutually exclusive classes: Positive (P), Negative (N) and Neutral (U)

- To attach sentiment scores to a tweet we leverage an Italian Sentiment Lexicon, namely a vocabulary whose lemmas are associated to pre-computed *positive* and *negative* sentiment scores

- Currently we are using the **Sentix** lexicon [Basile and Nissim 2013]

- Since it aligns several existing, independent lexical resources (*WordNet*, *MultiWordNet*, *BabelNet*, *SentiWordNet*) Sentix contains many duplicated lemmas
  - ~75'000 lemmas overall, only ~42'000 unique
    - ➡ To ensure unambiguous and reproducible results we de-duplicated Sentix by averaging atomic sentiment scores of duplicated lemmas

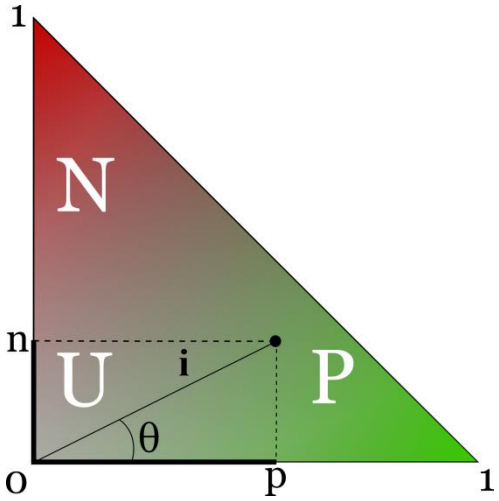
**Author: Diego Zardetto**

# The Sentiment Space

- In Sentix, positive (p) and negative (n) sentiment scores of lemmas are constrained as follows:

$$\begin{cases} p \in [0,1] \\ n \in [0,1] \\ p + n \leq 1 \end{cases}$$

- Therefore Sentix maps lemmas to points belonging to the sentiment triangle:




- From (p, n) coordinates we can pass to polar coordinates (i, θ) and derive two *additional* sentiment scores:
  - ✓ **Polarity** $\omega = 1 - 4\theta/\pi \quad \omega \in [-1, 1]$
  - ✓ **Intensity** $i = \sqrt{p^2 + n^2} \quad i \in [0, 1]$

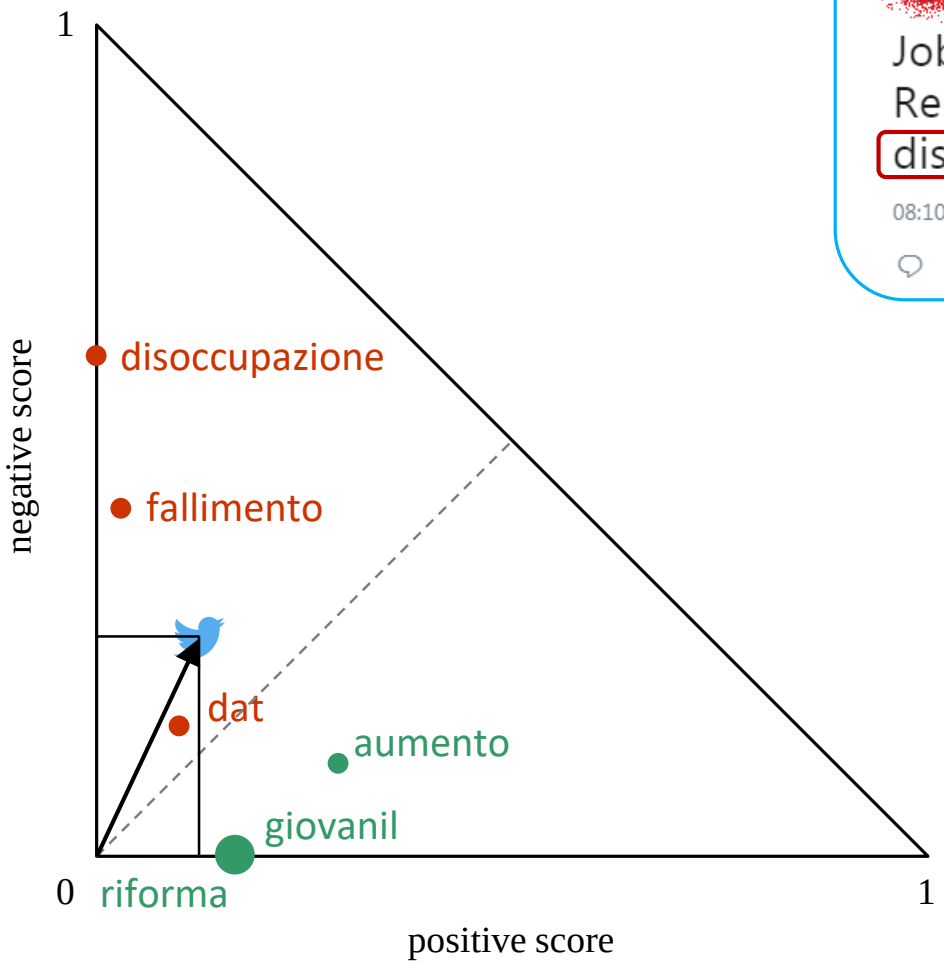
- This way Sentix lemmas are mapped to a 4D sentiment space

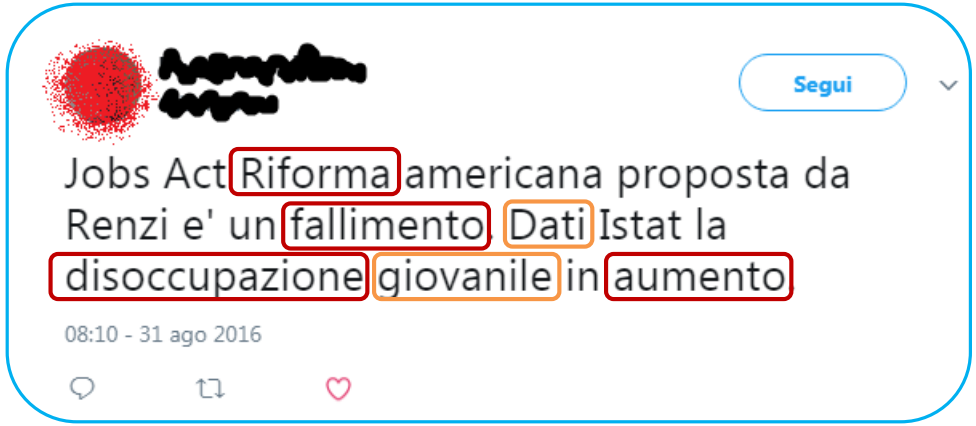
| lemma | pos | neg | polarity | intensity |
|---|---|---|---|---|
| caldo | 0.25 | 0.125 | 0.41 | 0.28 |
| freddo | 0.047 | 0.297 | -0.8 | 0.3 |

➡ To enable clustering, tweets too must be mapped to this 4D space

**Author: Diego Zardetto**

# From Word-level to Tweet-level Sentiment Scores



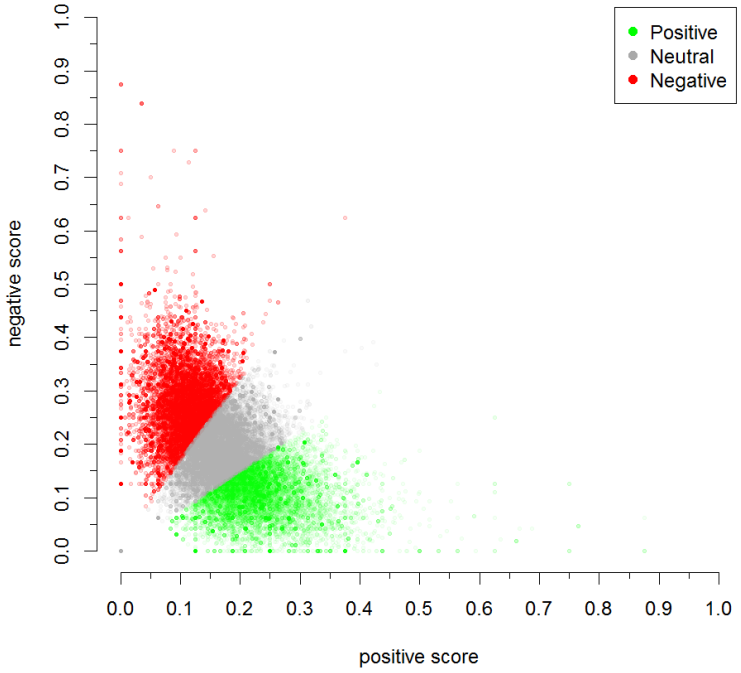

| word | pos | neg | polarity | intensity |
|---|---|---|---|---|
| riforma | 0.125 | 0 | 1 | 0.125 |
| fallimento | 0.021 | 0.375 | -0.929 | 0.376 |
| dat | 0.063 | 0.104 | -0.312 | 0.121 |
| disoccupazione | 0 | 0.625 | -1 | 0.625 |
| giovanil | 0.125 | 0 | 1 | 0.125 |
| aumento | 0.208 | 0.083 | 0.516 | 0.224 |

| pos | neg | polarity | intensity |
|---|---|---|---|
| 0.090 | 0.198 | -0.455 | 0.218 |

**Author: Diego Zardetto**

# Clustering and Calculation of the Index

- Once sentiment scores (p, n, ω, i) are available for all the tweets of a daily block…
- …we use K-means to cluster them into Positive, Negative and Neutral tweets
  - ✓ to lower the risk of finding a local optimum, we run it 100 times with random starts and pick the best solution
- Lastly we compute the daily index value (S), which depends on the distribution of tweets within the Positive, Neutral and Negative classes




$$S = \bar{\omega}_i = \frac{\sum_t i_t\ \omega_t}{\sum_t i_t} = \frac{\sum_{t \in P} i_t\ \omega_t + \sum_{t \in N} i_t\ \omega_t}{\sum_t i_t}$$

where $\omega_t \stackrel{\text{def}}{=} 0 \ \ \forall t \in \text{Neutral}$

- This index can be seen as the average of polarity (ω) weighted by intensity (i), provided we treat *Neutral* tweets as if their polarity were *zero*. Compared to traditional alternatives:
  - ✓ It is more resilient to tweets' misclassification
  - ✓ It reduces day-to-day volatility

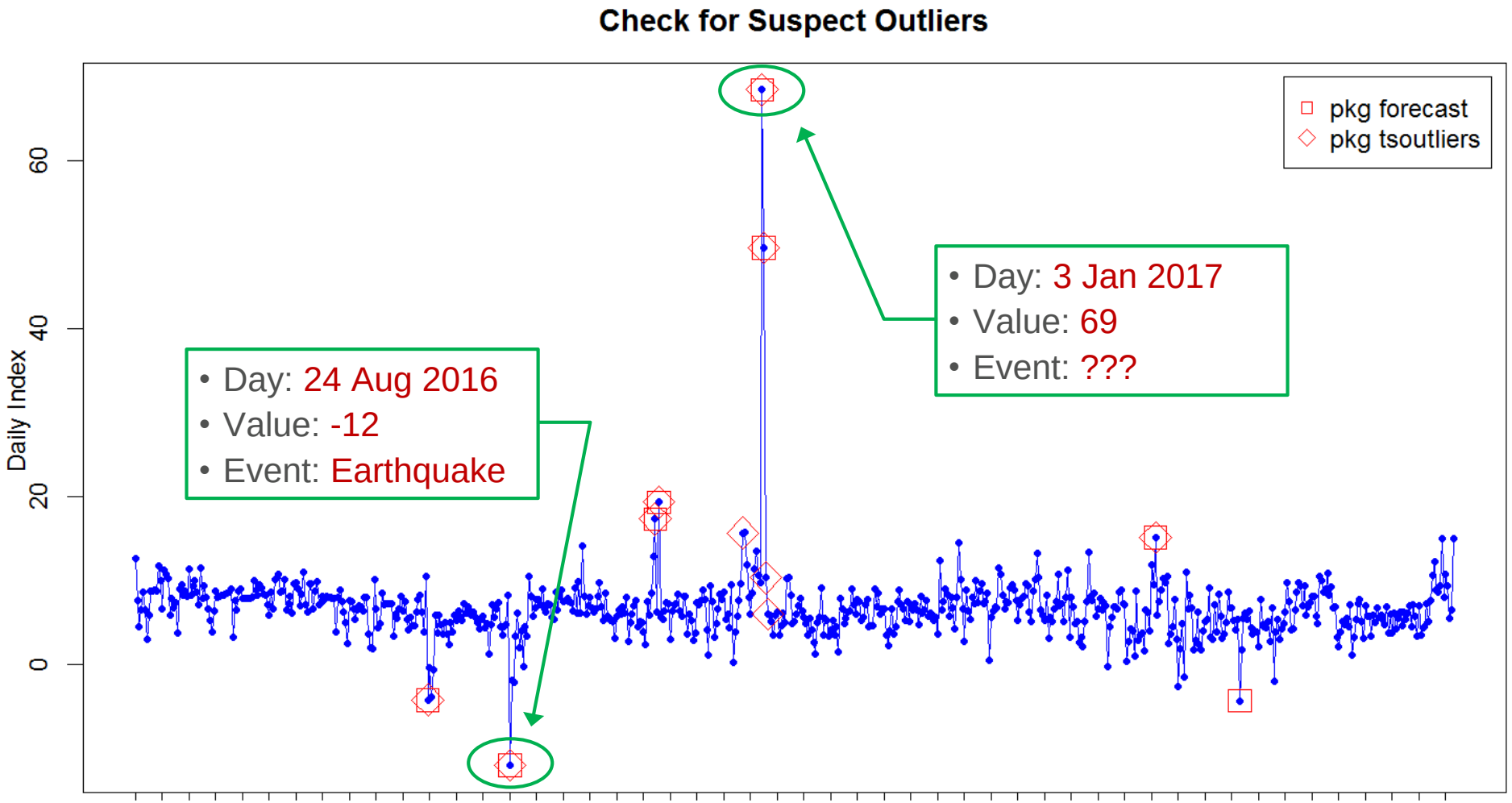
**Author: Diego Zardetto**

# Monitoring and Validation

- *No filter is perfect!* Thus we devoted special care to make the index robust against possible contaminations by off-topic tweets that might pass the filter
- We developed a surveillance system, which periodically searches for anomalous values in the daily time series by means of two independent and complementary outlier detection routines
  - Daily values detected as potential outliers cause the system to generate a set of automated diagnostic reports
  - These are then sent to human reviewers in charge of deciding whether the detected values are actually proper data points, or instead truly anomalous
- Truly anomalous data typically arise when an off-topic tweet that happened to pass the filter becomes **"viral"** on Twitter
  - Being re-twitted and quoted thousands of times in a day, viral tweets may have an unduly impact on the daily index and introduce bias
- All the daily index values classified as truly anomalous are eventually imputed via nearest-neighbor interpolation

**Author: Diego Zardetto**

# Anomalous Values: One Example




**Author: Diego Zardetto**

# 24 Aug 2016

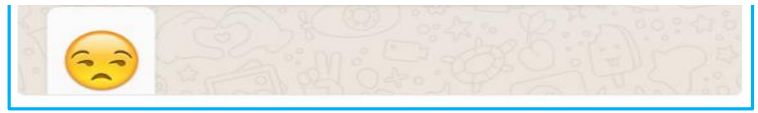

terremoto

provvisorio persone amatrice ultimora nadelparis italia sostegno morti macerie centro cuore almeno sisma dolore vicino prayforitaly accumoli colpite tragedia ansa famiglia sotto mercato bilancio vicini tutte italyquake vittime famiglie

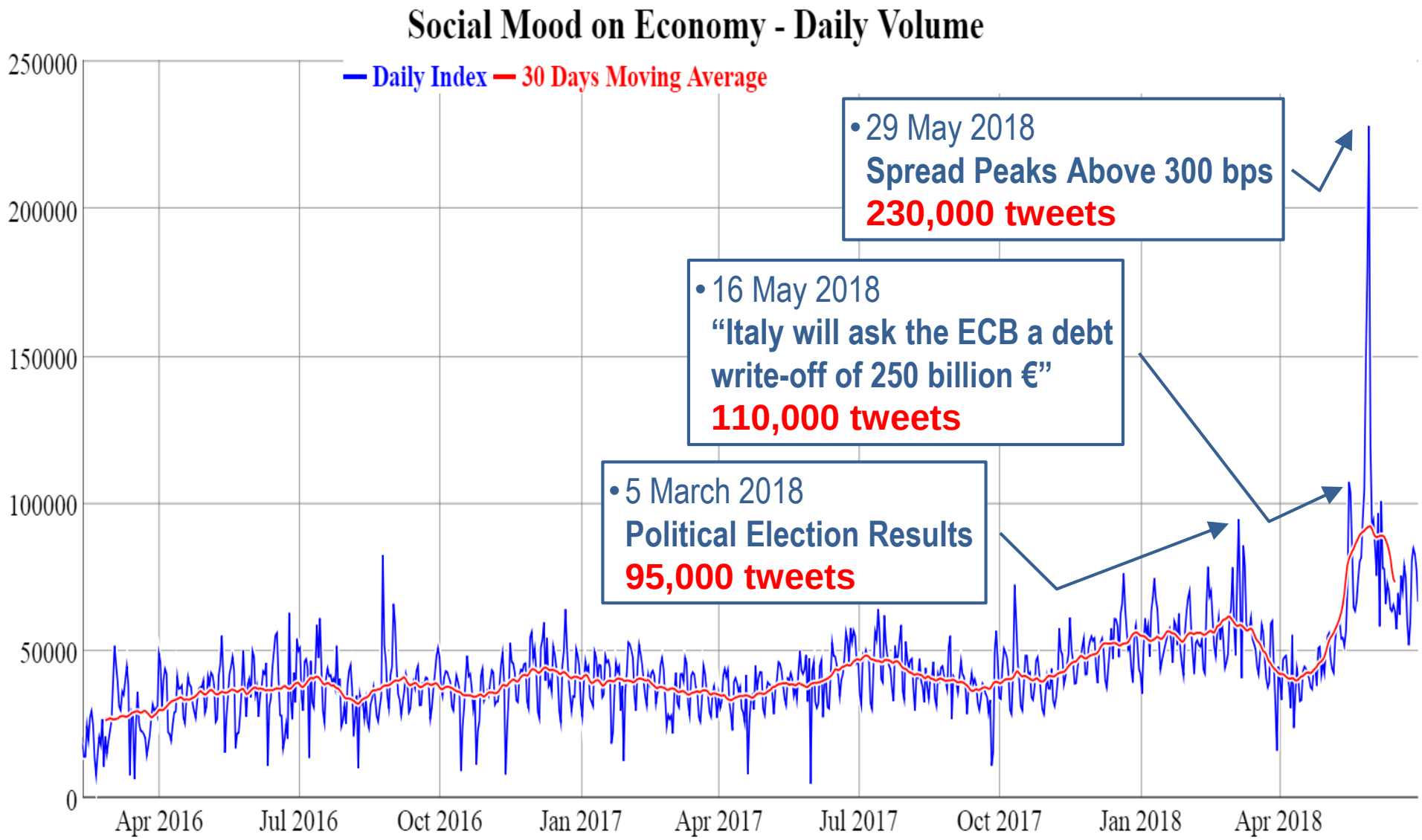
**Author: Diego Zardetto**

# 3 Jan 2017


prima fedefederossi ben spesa

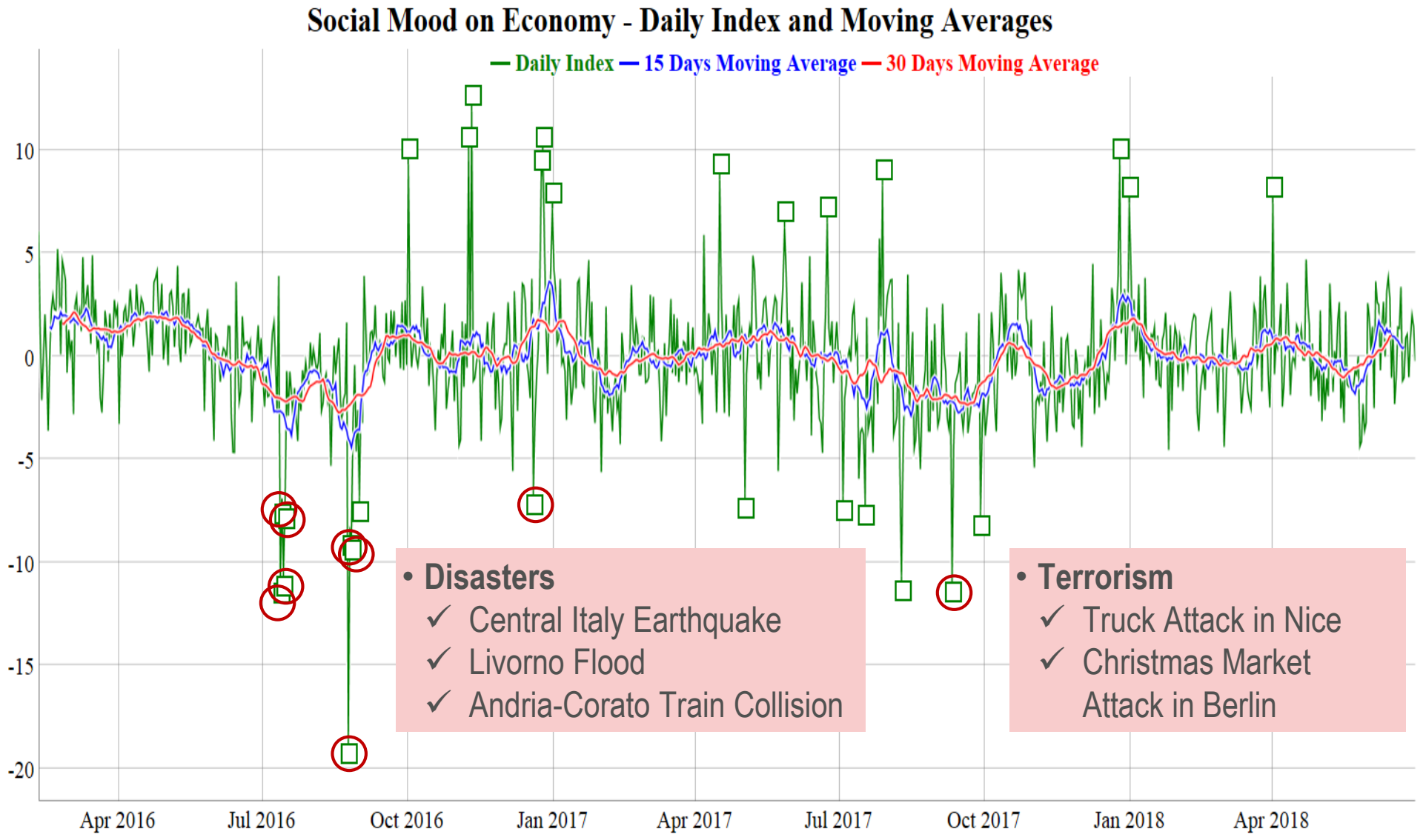


# Volume Burst in 2018 Post-Election Crisis



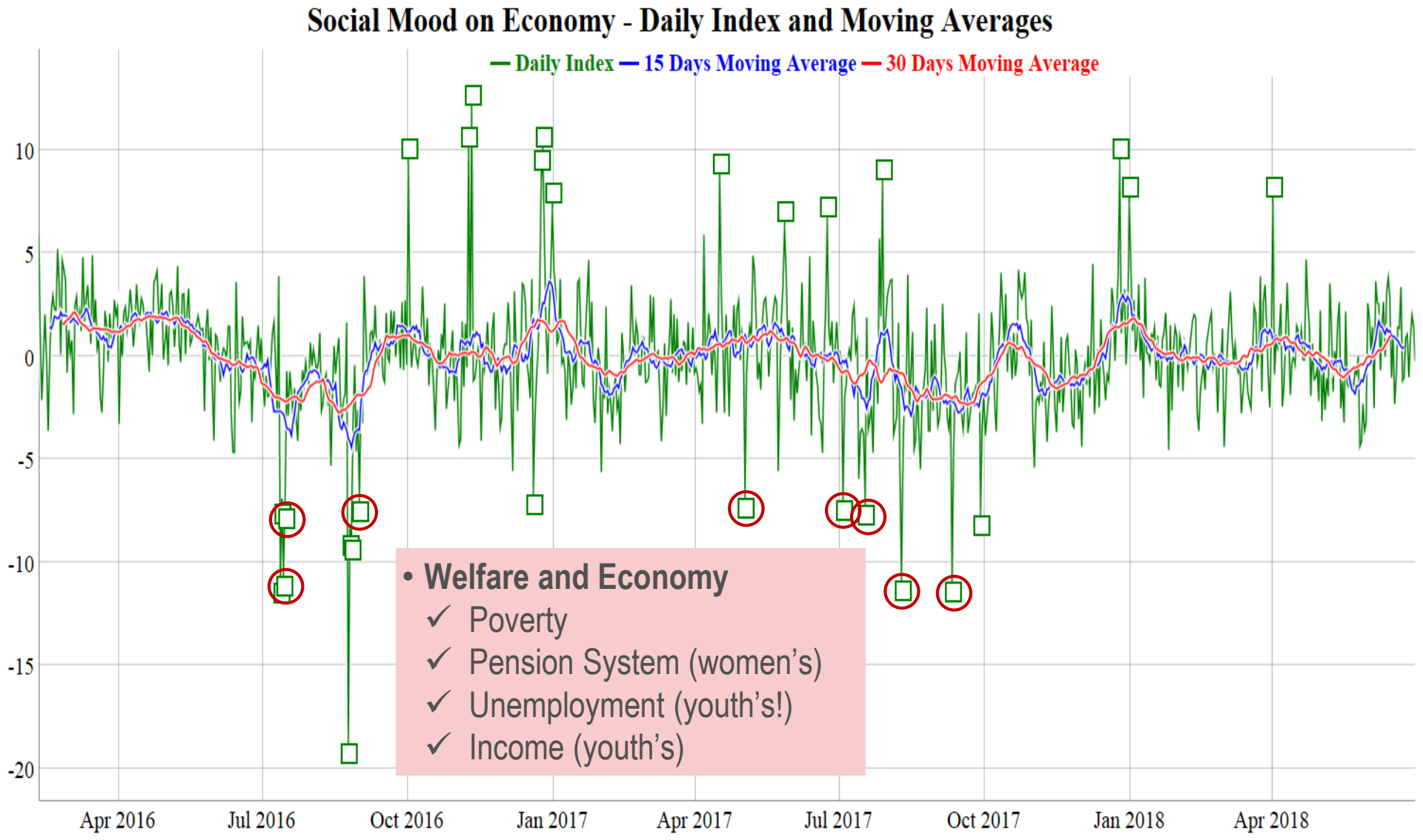

**Author: Diego Zardetto**

# Valleys: Disasters and Terrorism




**Author: Diego Zardetto**

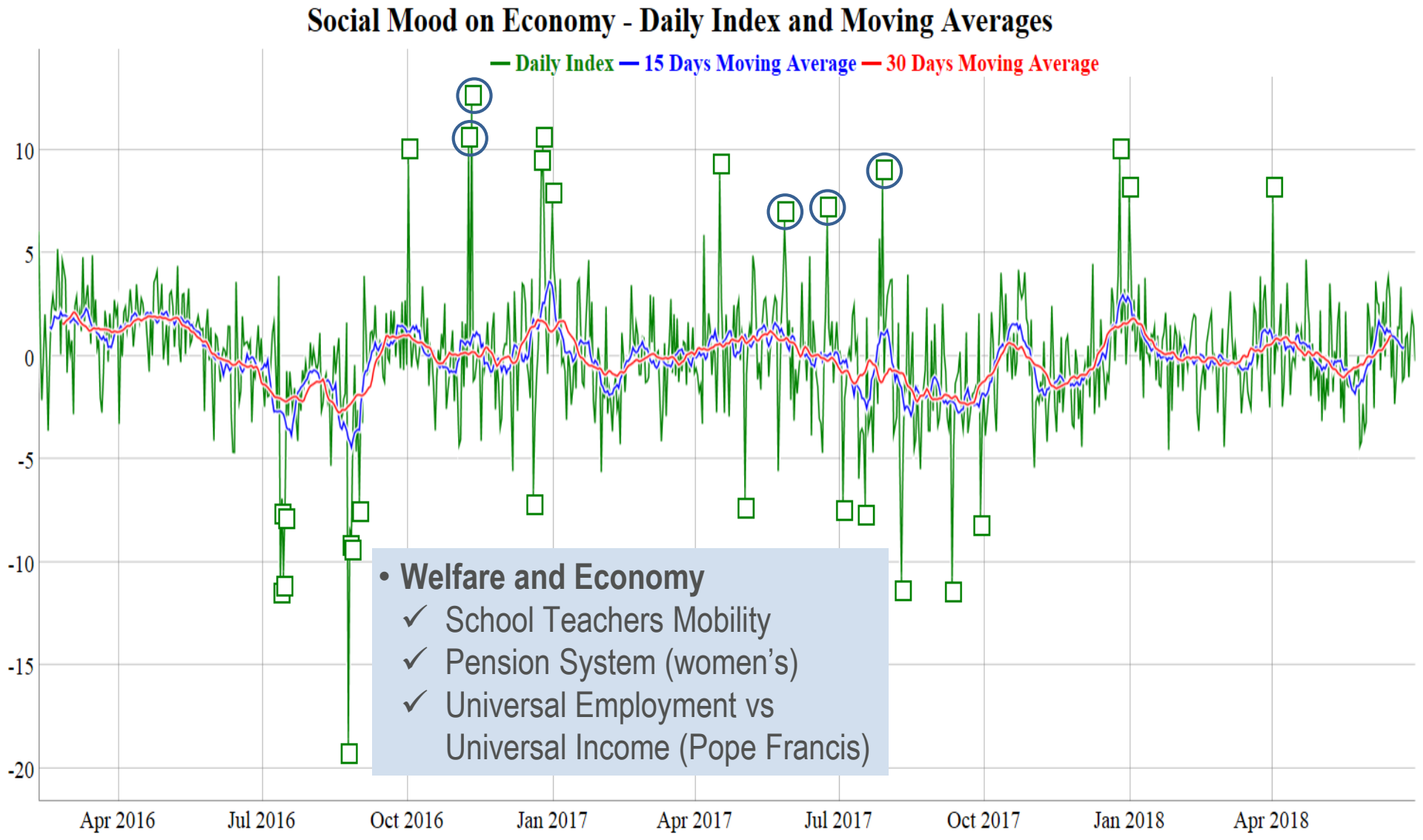
# Valleys: Welfare and Economy




- **Welfare and Economy**
  - ✓ Poverty
  - ✓ Pension System (women's)
  - ✓ Unemployment (youth's!)
  - ✓ Income (youth's)

**Author: Diego Zardetto**

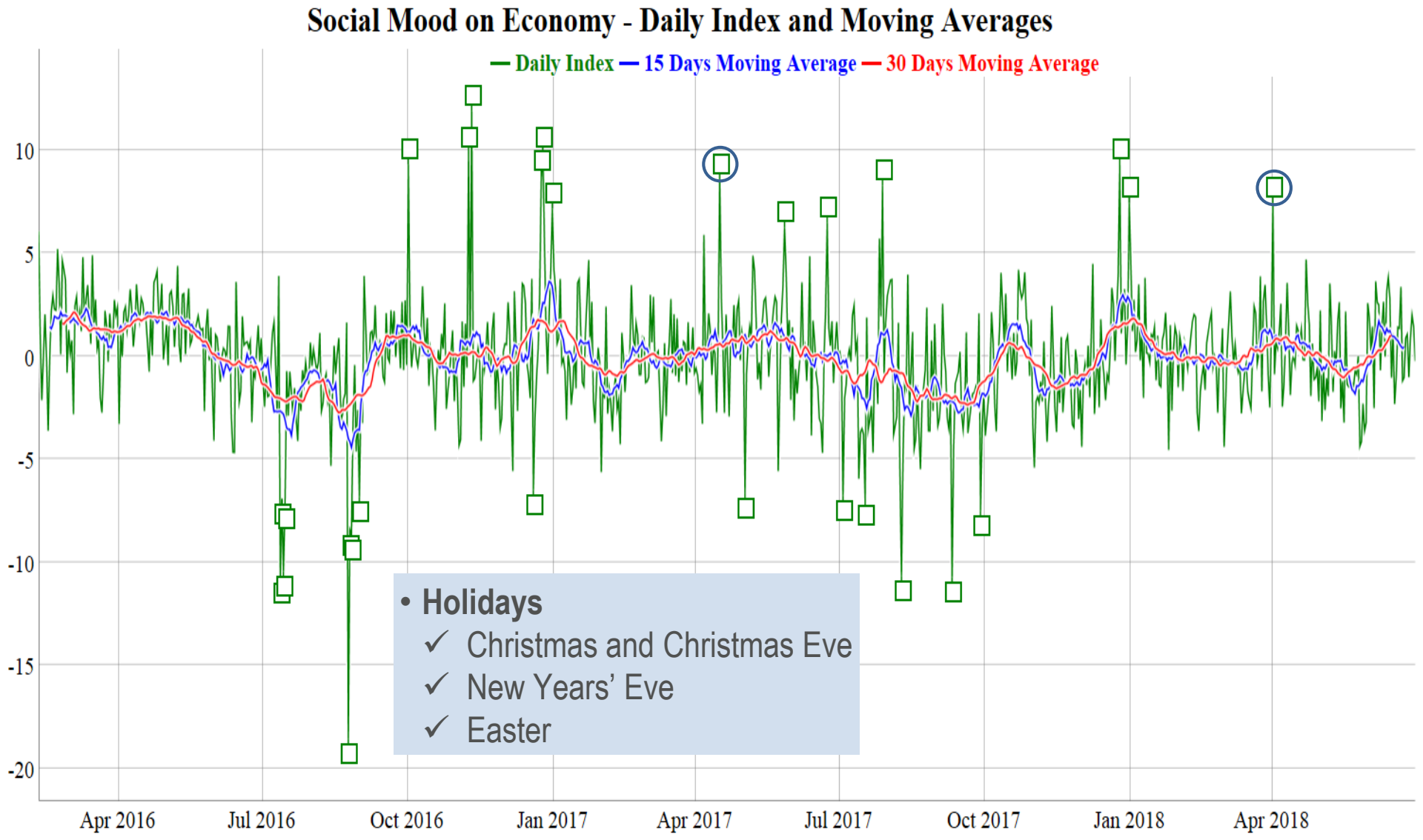
# Peaks: Welfare and Economy




- **Welfare and Economy**
  - ✓ School Teachers Mobility
  - ✓ Pension System (women's)
  - ✓ Universal Employment vs Universal Income (Pope Francis)

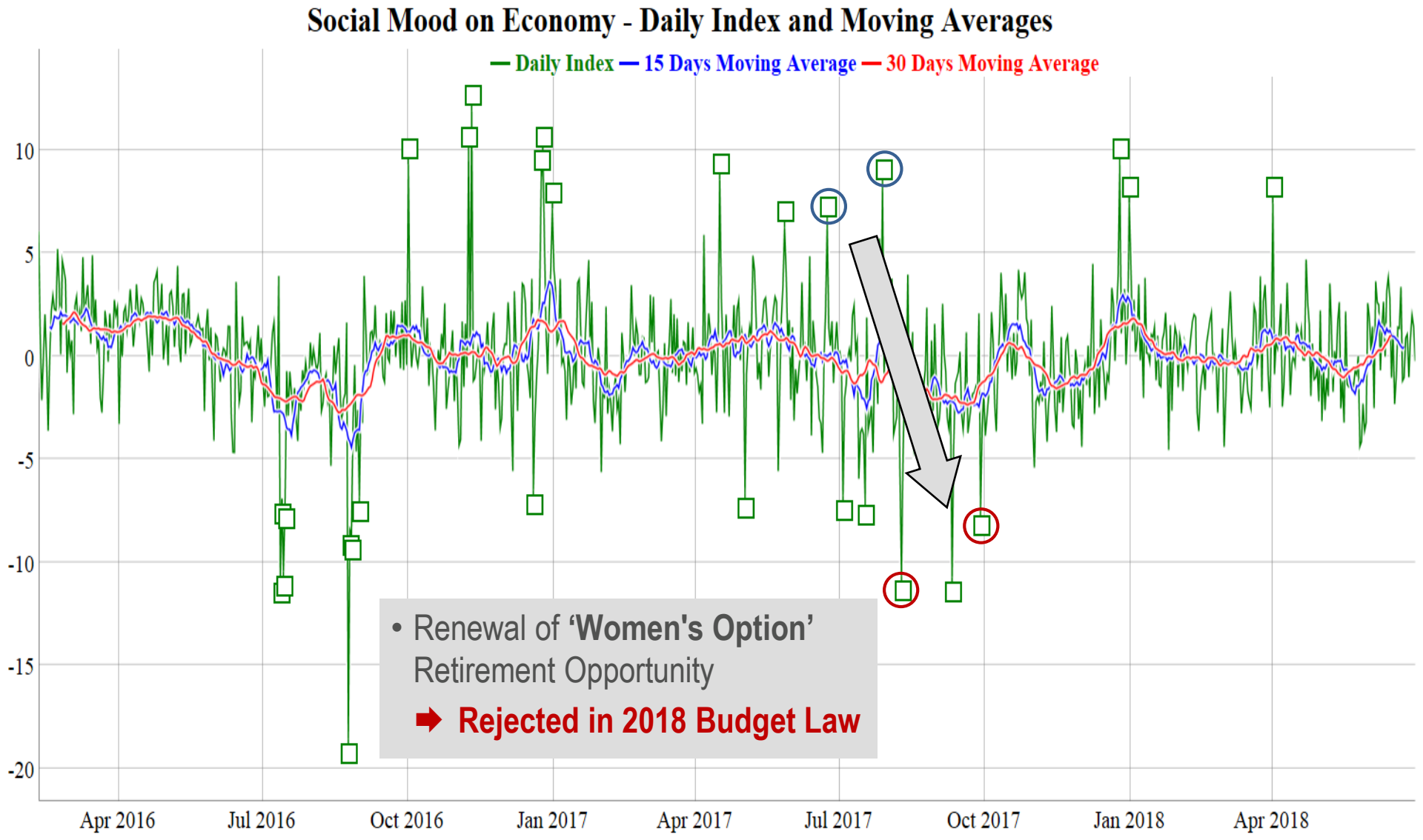
**Author: Diego Zardetto**

# Peaks: Holidays




**Author: Diego Zardetto**

# An Interesting Dynamic

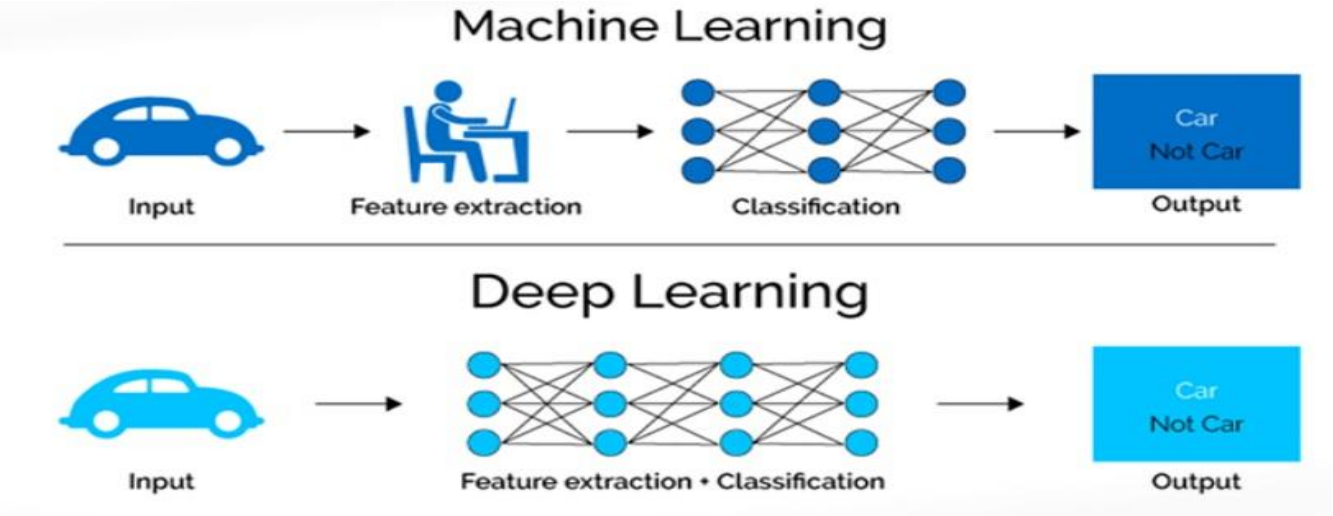



**Author: Diego Zardetto**

# What is Deep Learning ?

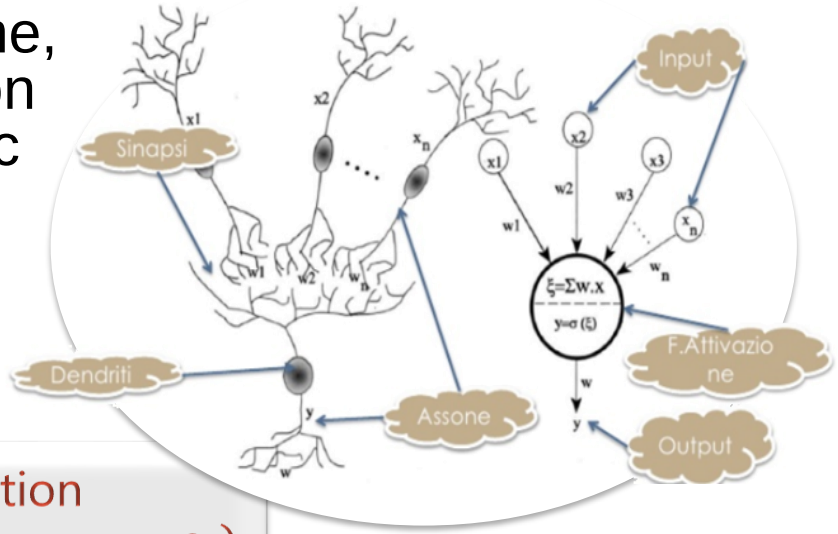
Deep Learning refers to algorithms that automatically ‘model’ high-level abstractions in data

i. here ‘model’ means: define, find, recognize and exploit
ii. here ‘automatically’ means: directly from data, without hinging upon handcrafted, task-specific features.



# ARTIFICIAL NEURAL NETWORKS (ANNs)

**ANNs** were introduced, for the first time, by 1943, in a work on the formalization of neural activity in propositional logic form (*McCulloch & Pitts, 1943*).
**We can define ANNs as a simple model of biological organisms' nervous system**.




Neuron Activation

$$A_j = \sum_{i=1}^{N} w_{ij} X_i - \theta_i$$

Activation function

$$y_j = \Phi(A_j) = \Phi\left(\sum_{i=1}^{N} w_{ij} X_i - \theta_i\right)$$

**In data mining:** Methods have been developed to produce comprehensible models and reduce training times:
**1)Rule extraction**: extraction of symbolic models from pre-trained neural networks.
**2)Learn simple**, easy-to-understand neural networks.

# DEEP LEARNING: Neural Networks become more effective

In recent years **Deep Neural Networks** have achieved noticeably breakthroughs in research (*Bengio, 2009*). This new methodology dealing with deep neural networks and their training algorithms is called *"Deep Learning"*. **So far, in all the experiments, the resulting performances were many magnitudes better than other machine learning techniques available.**

GOOGLE DATACENTER

| | |
|---|---|
| 1,000 CPU Servers<br>2,000 CPUs • 16,000 cores | 600 kWatts<br>$5,000,000 |

STANFORD AI LAB

| | |
|---|---|
| 3 GPU-Accelerated Servers<br>12 GPUs • 18,432 cores | 4 kWatts<br>$33,000 |

# DEEP LEARNING: a cutting-edge approach to Computer Vision and NLP

**GPU**

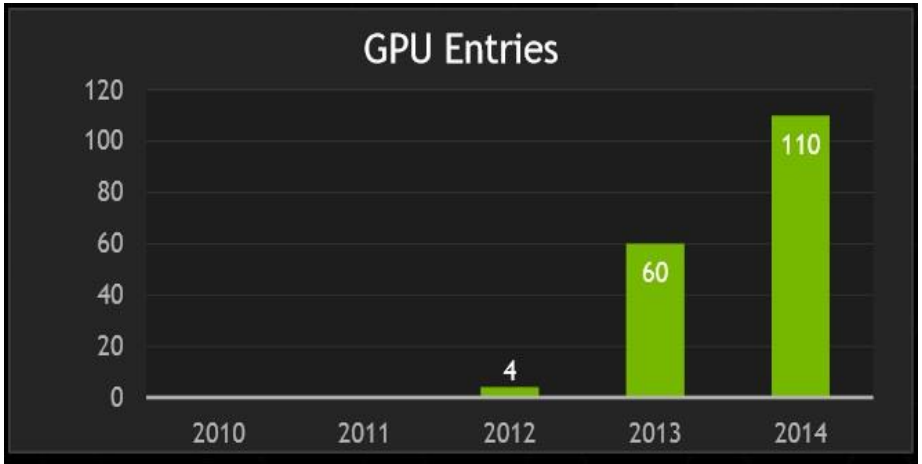
- The advent of GPUs makes possible the training of very large neural networks with even more than 150 millions of parameters.

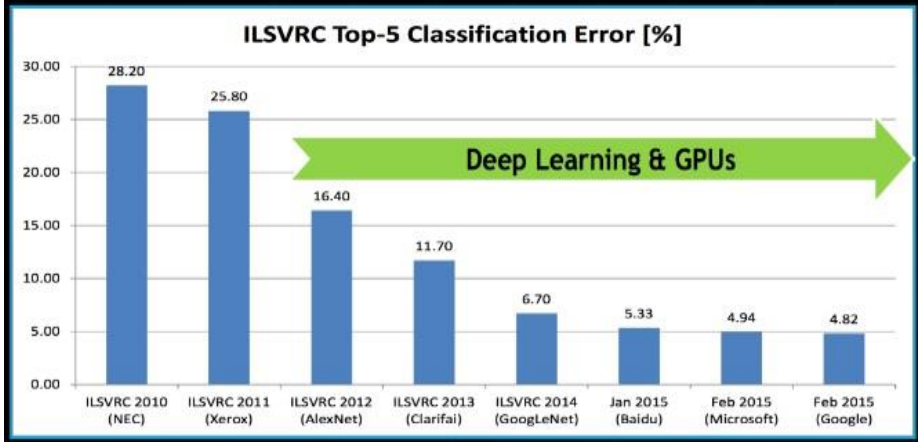
**BIG DATA**

- A new generation of larger training and test sets.

**DROPOUT**

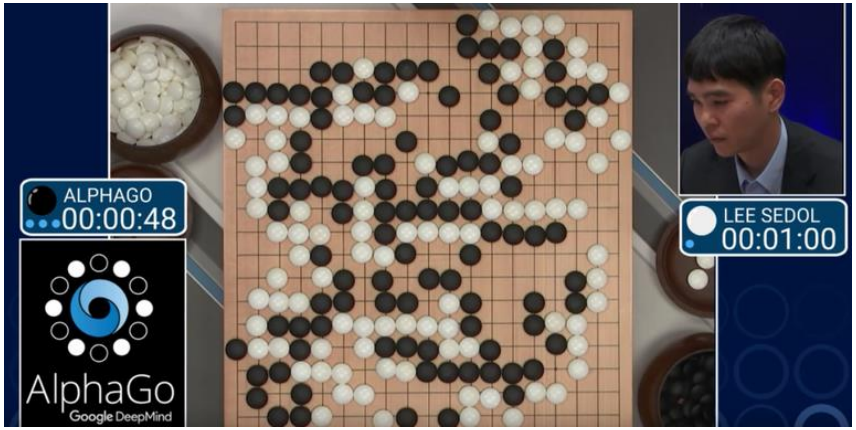
- Better model regularization techniques have been discovered such as ***"Dropout"*** or ***"Data Augmentation"***



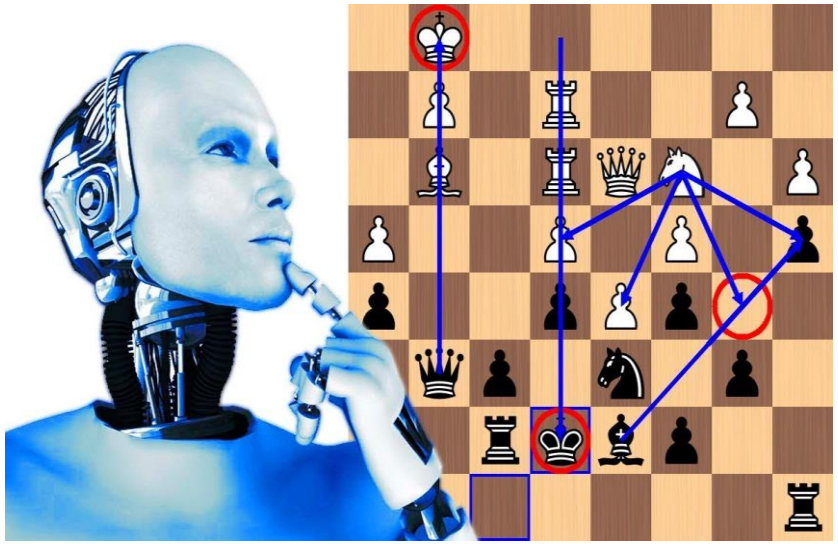
# Why Deep Learning over-performed traditional statistics models?

- "Deep Learning" approaches can be **end-to-end trained** without a task-specific feature engineering.
- **These model are scalable**: adding GPUs they can be trained faster.
- **"Deep Learning is killing every problem in AI"** (*Elizabeth Gibney, 2016*)
- **Basically, statistics is not able to deal with very high dimensionalities of data as Deep Learning does**.



# Alpha Zero: Mastering the games of Go and Chess without Human Knowledge

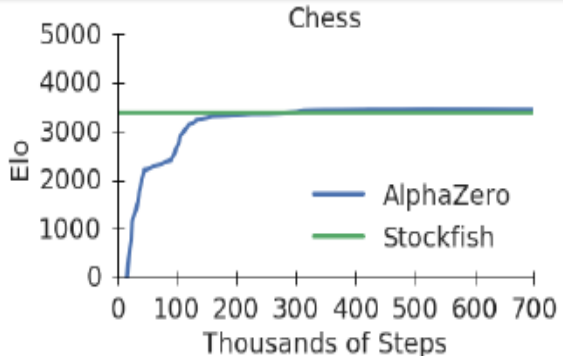
- **In Just 4 Hours, Google's AI Mastered All The Chess Knowledge in History**
- **"I always wondered how it would be if a superior species landed on Earth and showed us how they played chess. Now I know." grandmaster Peter Heine Nielsen.**

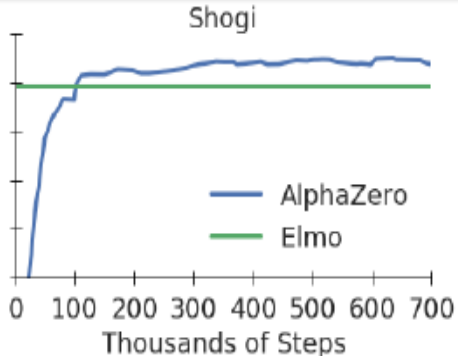


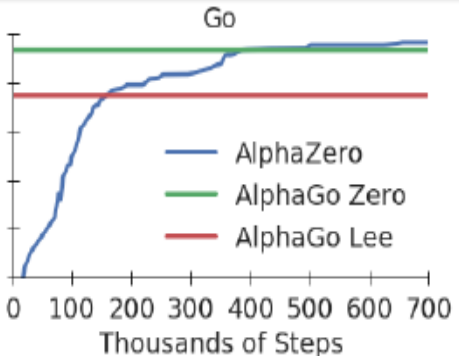
- **Google's AlphaZero Destroys Stockfish In 60 Game Matches**

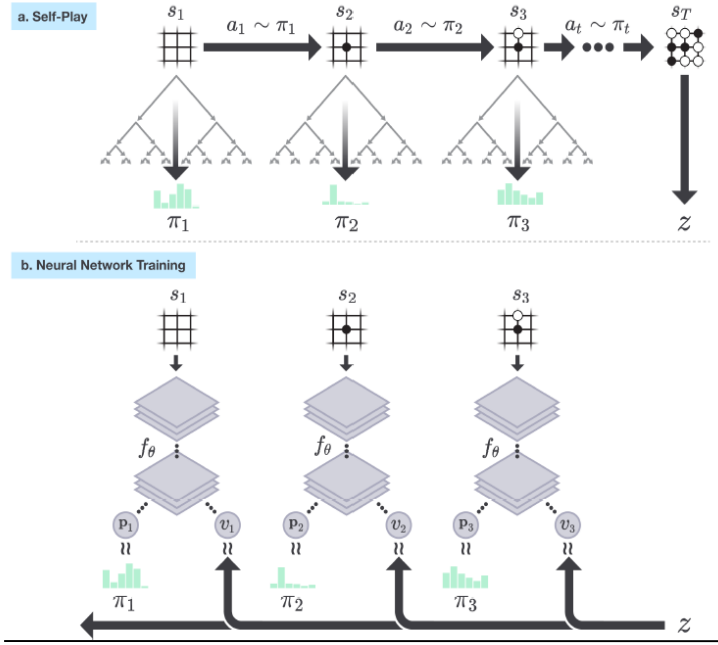
**"This algorithm could run cities, continents, universes."**
***PETER DOCKRILL* (*Senior Writer*)**

# Alpha Zero IS an Artificial Intelligence, it IS NOT just a Chess Engine..

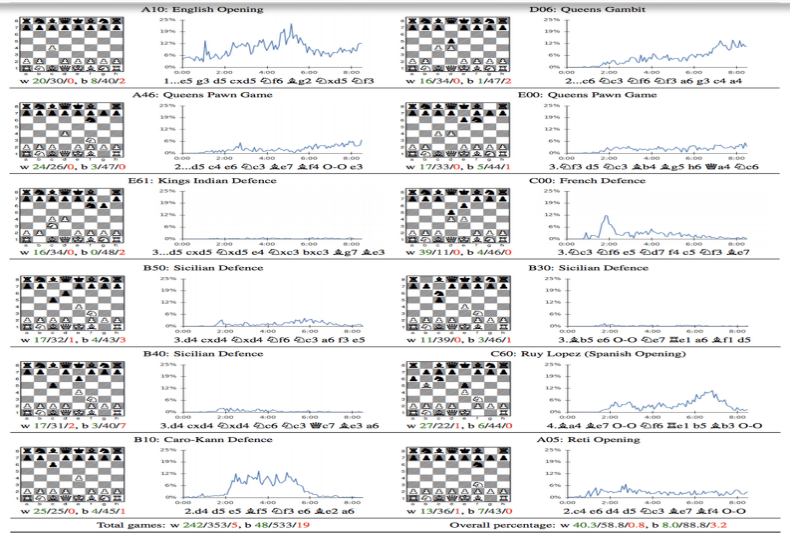


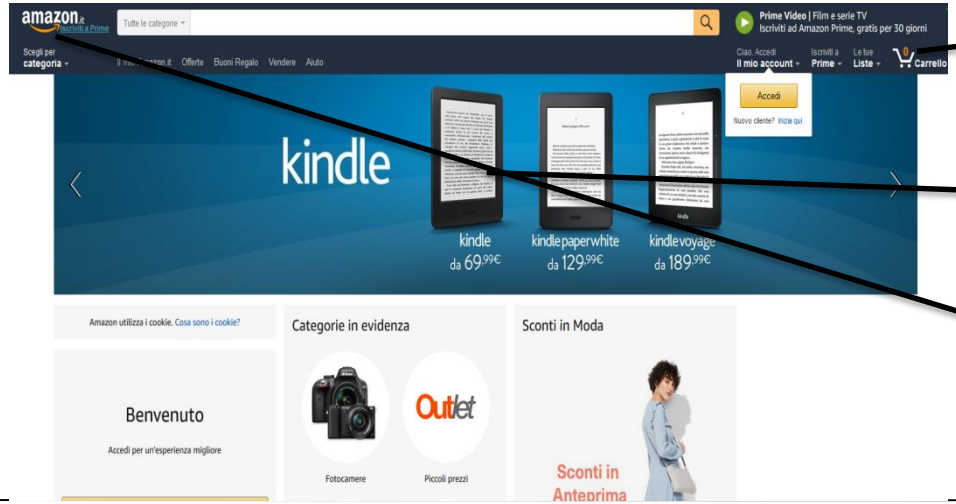




## 12 Chess Openings Discovered by Alphazero



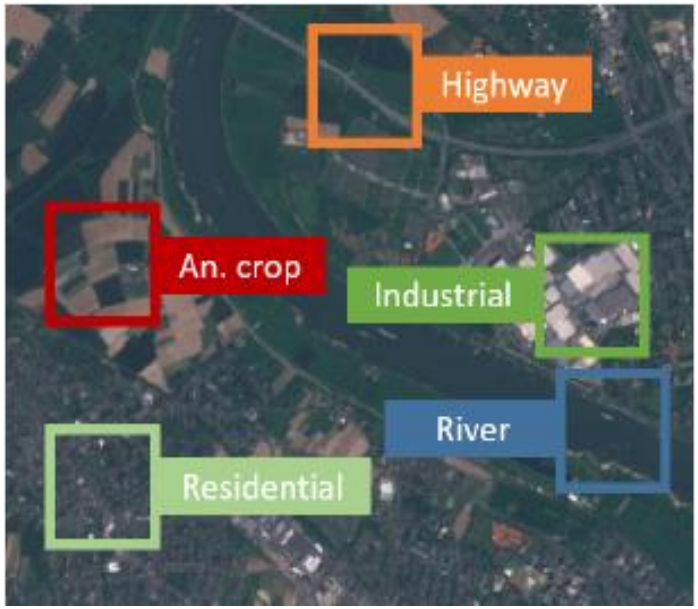
# Web-Site classification by Images Approach

- According to the **False Positive Reduction** technique we exploit the inner images segmentation of a Web-site in order to train an evolved ConvNet (ResNet) model onto the single websites images segments.
- **ConvNet** is trained in **“Transfer Learning”** mode, which means taking advantage of a pre-trained model onto well-know datasets such as Imagenet (1000 image classes, 1.2 mln images)




# Automatic Extraction of Statistics from Satellite Imagery: Land Use and Land Cover Classification (Helber, et al., )

Nowadays, more and more public and up-to-dated **satellite image** data for Earth observation are available.



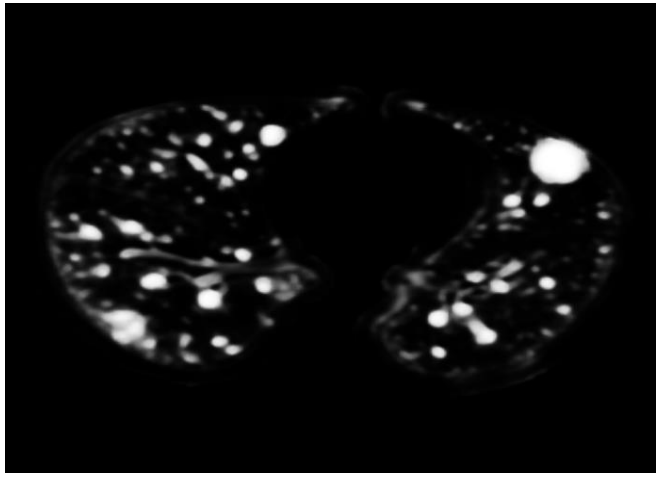

However, to fully utilize this data, to automatically extract statistics, satellite images must be processed and transformed into structured semantics.

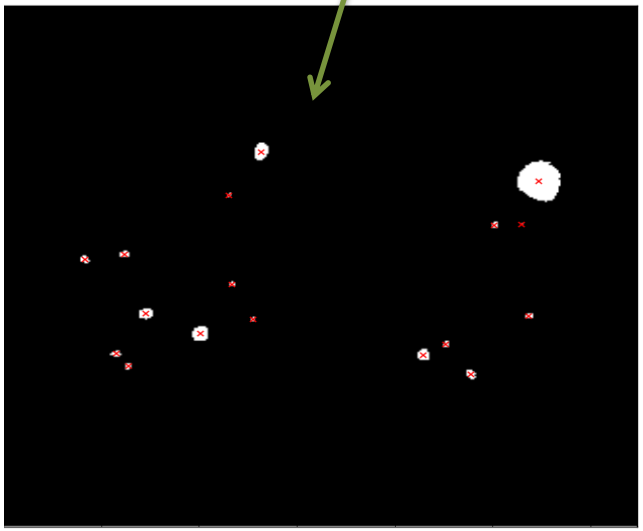
# Lung Cancer Classification

Candidate Nodule Selection via UNET

Dilation, Erosion, Nodules Distance Merging

False Positive Reduction via WideResNet





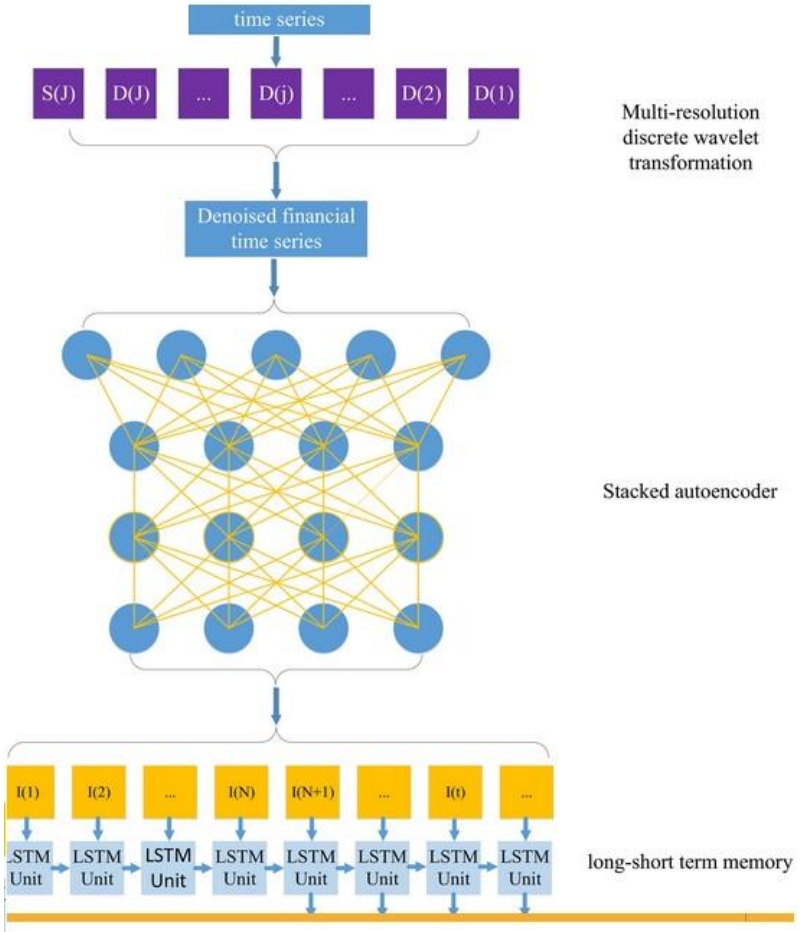
Cancer / Non cancer

# Advantages of Artificial Neural Networks (ANNs) in Time-Series Prediction

- However, by using ANNs, a priori analysis as ANNs do not require prior knowledge of the time series structure because of their black-box properties **(Nourani, et al., 2009)**.

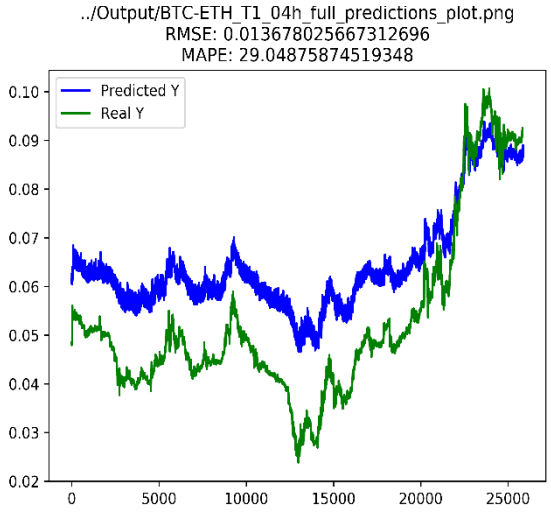
- Also, the impact of the stationarity of time series on the prediction power of ANNs is quite small. It is feasible to relax the stationarity condition to non-stationary time series when applying ANNs to predictions **(Kim, et al., 2004)**.

- ANNs allow **multivariate time-series forecasting** whereas classical linear methods can be difficult to adapt to multivariate or multiple input forecasting problems.

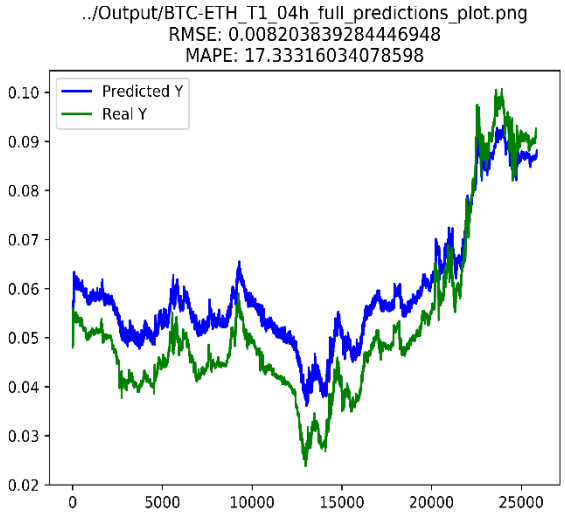


# Results: Bitcoin BTC-ETH exchange Time Series Prediction – 4 hours (Poloniex)

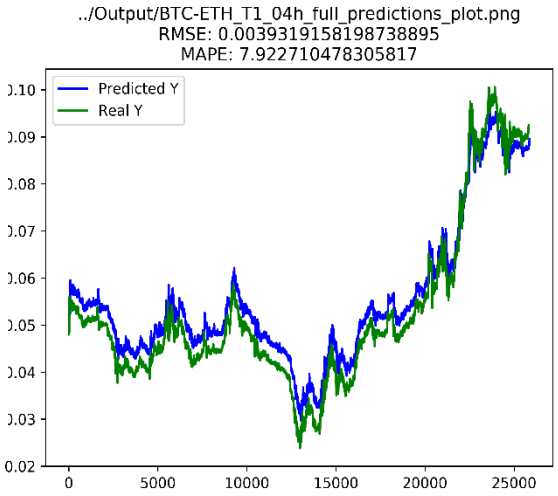
## Test Set : 10%




1 Epoch



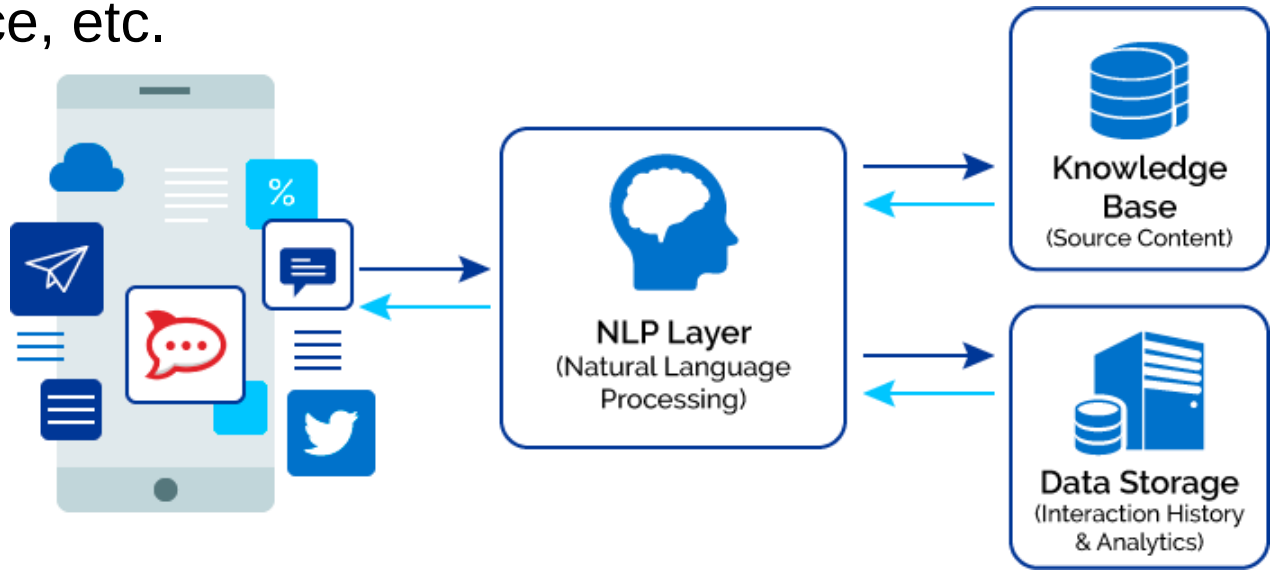

10 Epochs




100 Epochs

# Textual Big Data alias The problem of the Natural Languale Processing - NLP

- Understanding **complex language utterances is one of the hardest challenge** for Artificial Intelligence (AI) and Machine Learning (ML).
- **NLP** is everywhere because people communicate most everything: web search, advertisement, emails, customer service, etc.



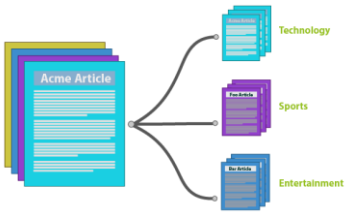
# Deep Learning and NLP

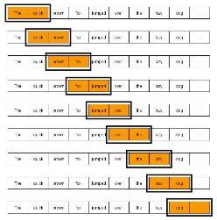
- “Deep Learning” approaches have obtained very high performance across many different **NLP** tasks. These models can often be trained with a **single end-to-end model** and do not require traditional, task-specific feature engineering.
  (*Stanford University School Of Engineering – CS224D*)

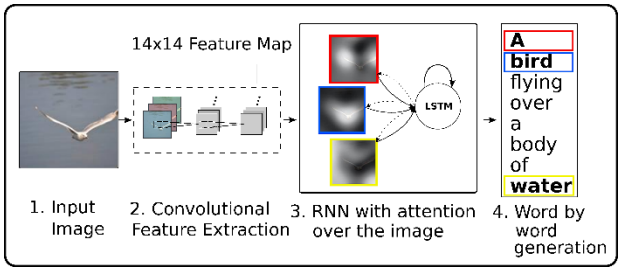
- **Natural language processing** is shifting from statistical methods to **Neural Networks**.

# 7 NLP applications where Deep Learning achieved «state-of-art» performance

- **1 <u>Text Classification</u>**: Classifying the topic or theme of a document (i.e. **Sentiment <u>Analysis</u>**).
- **2 <u>Language Modeling</u>**: Predict the **next word given the previous words**. It is fundamental for other tasks.
- **3 <u>Speech Recognition</u>**: Mapping an **acoustic signal** containing a spoken natural language utterance into the corresponding sequence of words intended by the speaker.
- **4 <u>Caption Generation</u>**: Given a **digital image**, such as a photo, generate a **textual description** of the contents of the image.



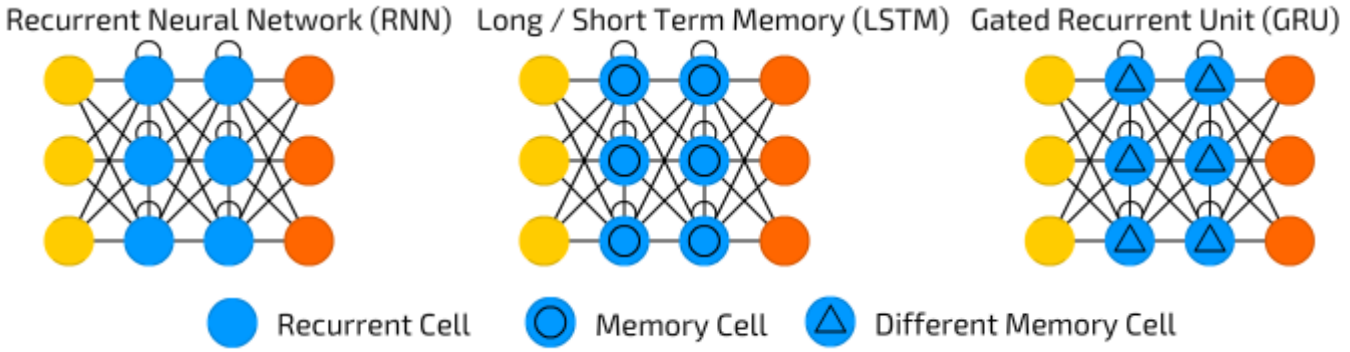


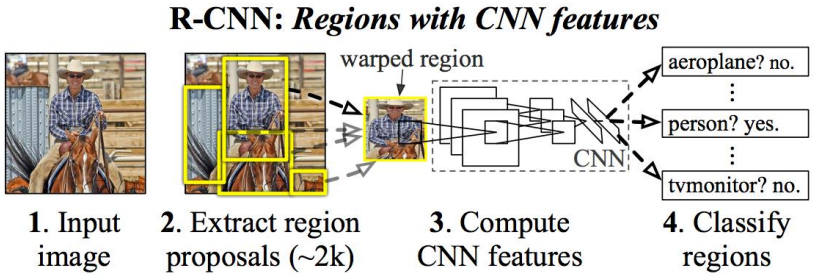
# 7 NLP applications where Deep Learning achieved «state-of-art» performance

- **5 Machine Translation**: Automatic translation of text or speech from one language to another, is one [of] the most important applications of NLP.
- **6 Document Summarization**: It is the task where a short description of a text document is created.
- **7 Question Answering**: It is the task where the system tries to answer a user query that is formulated in the form of a question by returning the appropriate noun phrase such as a location, a person, or a date. (i.e. Who killed President Kennedy? Oswald)

# Text Classification Models

- **RNN, LSTM, GRU, ConvLstm, RecursiveNN, RNTN, RCNN**
- The modus operandi for text classification involves the use of a pre-trained **word embedding** for **representing words** and a **deep neural networks** for **learning how to discriminate documents** on classification problems.




- The **non-linearity of the NN** leads to superior classification accuracy.

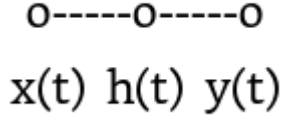
# Recurrent Neural Networks (RNNs) : Elman's Architecture

- There exist seveal indicators to measure the predictive accuracy of each model (**Hsieh, et. al.,** 2011; **Theil**, 1973)

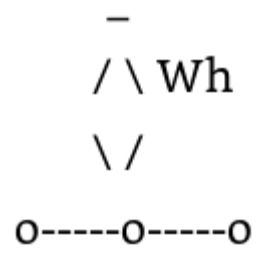
- **RMSE (Root Mean Square Error): R**epresents the sample standard deviation of the differences between predicted values and observed values.
- **MAPE (Mean Absolute Percentage Error):** Measures the size of the error in percentage terms. Most people are comfortable thinking in percentage terms, making the MAPE easy to interpret.
- Thanks to its recursive formulation, RNNs are not limited by the **Markov assumption** for sequence modeling:

$$p\{ x(t) \mid x(t-1), \ldots, x(1) \} = p\{ x(t) \mid x(t-1) \}$$

**Simple Feed Forward Artificial Neural Network (MLP)**




**Recurrent Neural Network (Elman's Architecture)**




$$h(t) = f(x(t)W_i + h(t-1)W_h + b_h)$$

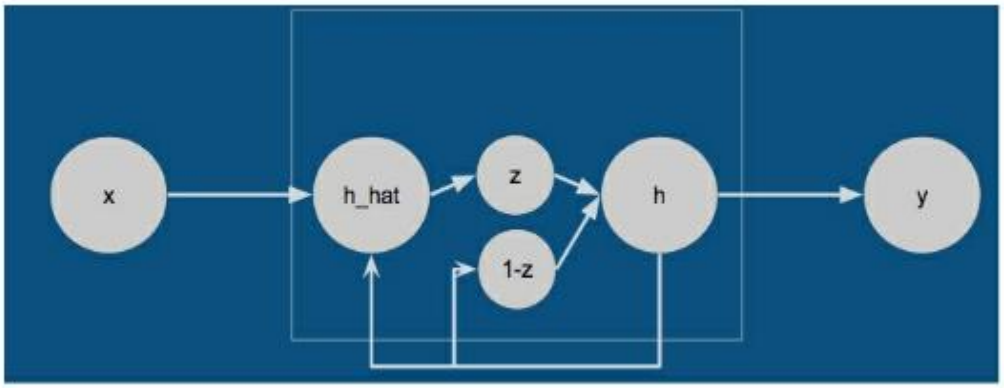
# Rated Recurrent Neural Networks (RRNNs)

- The idea is to weight *f(x, h(t-1)),* which is the output of a simple RNN and *h(t-1)* which is the previous state **(Amari, et al., 1995)**.

- **We add a rating operation between what would have been the output of a simple RNN and the previous output value.**

- **This new operation can be seen as a gate since it takes a value between 0 an 1, and the other gate has to take 1 minus that value**

- **This is a gate that is choosing between 2 things: a) taking on the old value or taking the new value. As result we get a mixture of both.**




$$h\_hat(t) = f(x(t)W_x + h(t-1)W_h + b_h)$$

$$z(t) = \text{sigmoid}(x(t)W_{xz} + h(t-1)W_{hz} + b_z)$$

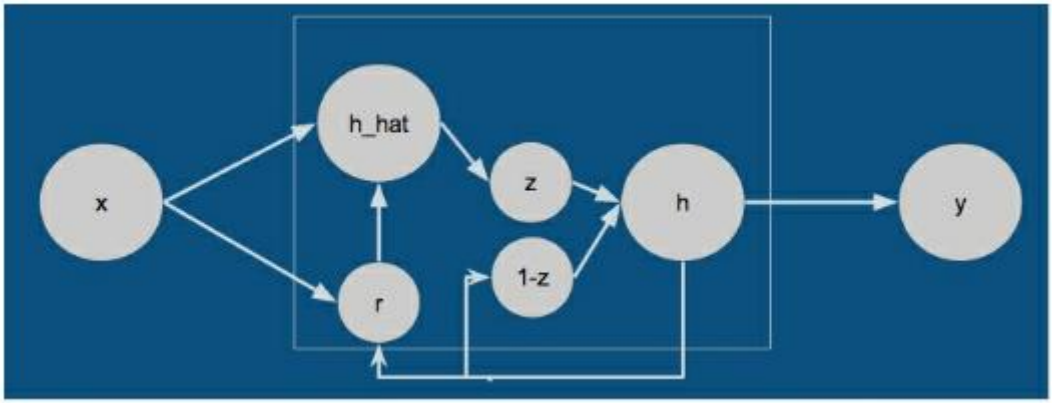
$$h(t) = (1 - z(t)) * h(t-1) + z(t) * h\_hat(t)$$

- **Z(t) is called the "rate"**

# Gated Recurrent Neural Networks (GRUs)

- Gated Reccurrent Units were introduced in 2014 and are a simpler versione of LSTM. They have less parameters but same concepts **(Chung, et al., 2014).**

- **Recent research has also show that the accuracy between LSTM and GRU is comparable and even better with the GRUs in some cases.**
- **In GRUs we add one more gate with regard to RRNNs: the "reset gate r(t)" controlling how much of the previous hidden we will consider when we create a new candidate hidden value. In other words, it can "reset" the hidden value.**
- **The old gate of RRNNs is now called "update gate z(t)" balancing previous hidden values and new candidate hidden value for the new hidden value.**

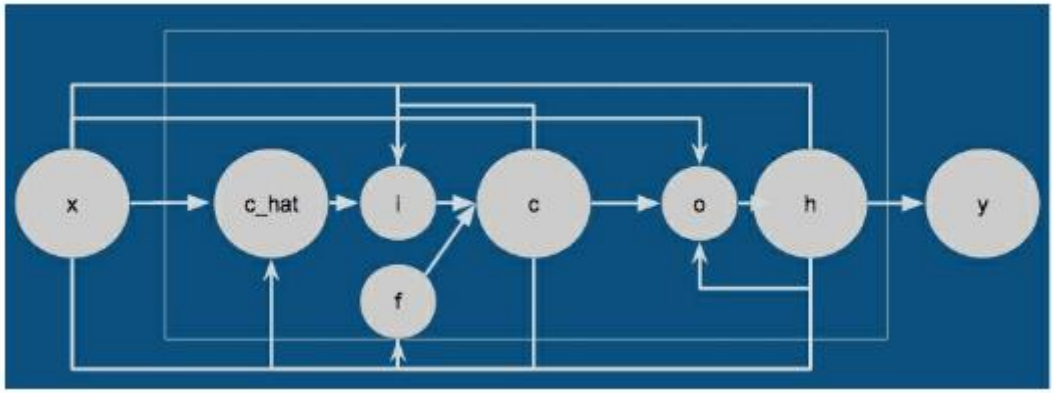



$$r_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r)$$
$$z_t = \sigma(x_t W_{xz} + h_{t-1} W_{hz} + b_z)$$
$$\hat{h}_t = g(x_t W_{xh} + (r_t \odot h_{t-1}) W_{hh} + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t.$$

# Long-Short Term Memories (LSTMs)

- LSTM is an effective solution for combating vanishing gradients by using memory cells **(Hochreiter, et al., 1997).**

- **A memory cell is composed of four units: an input gate, an output gate, a forget gate and a self-recurrent neuron**
- **The gates control the interactions between neighboring memory cells and the memory cell itself. Whether the input signal can alter the state of the memory cell is controlled by the input gate. On the other hand, the output gate can control the state of the memory cell on whether it can alter the state of other memory cell. In addition, the forget gate can choose to remember or forget its previous state.**

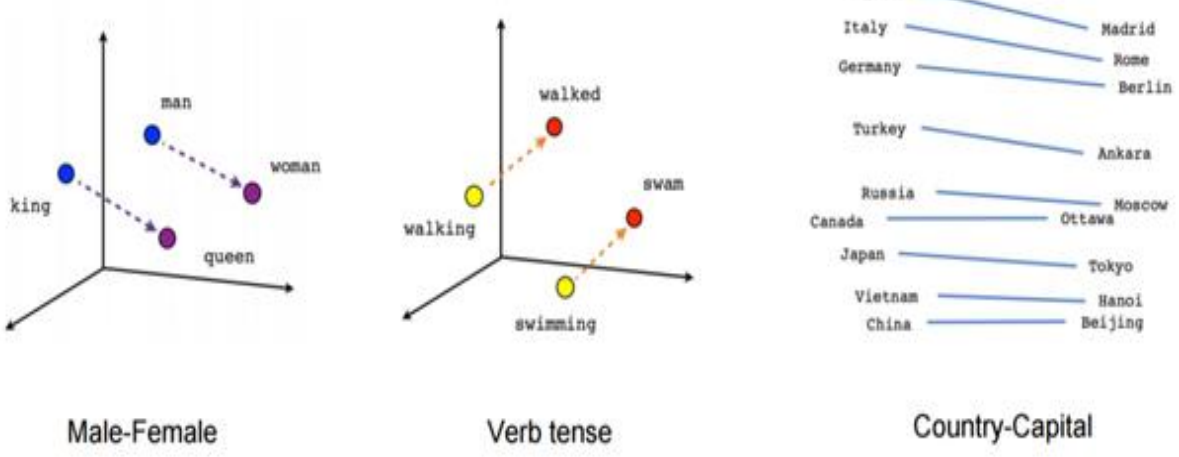



$$i_t = \sigma(x_t W_{xi} + h_{t-1} W_{hi} + c_{t-1} W_{ci} + b_i)$$
$$f_t = \sigma(x_t W_{xf} + h_{t-1} W_{hf} + c_{t-1} W_{cf} + b_f)$$
$$c_t = f_t c_{t-1} + i_t \tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c)$$
$$o_t = \sigma(x_t W_{xo} + h_{t-1} W_{ho} + c_t W_{co} + b_o)$$
$$h_t = o_t \tanh(c_t)$$

# Word Embedding & Language Modeling

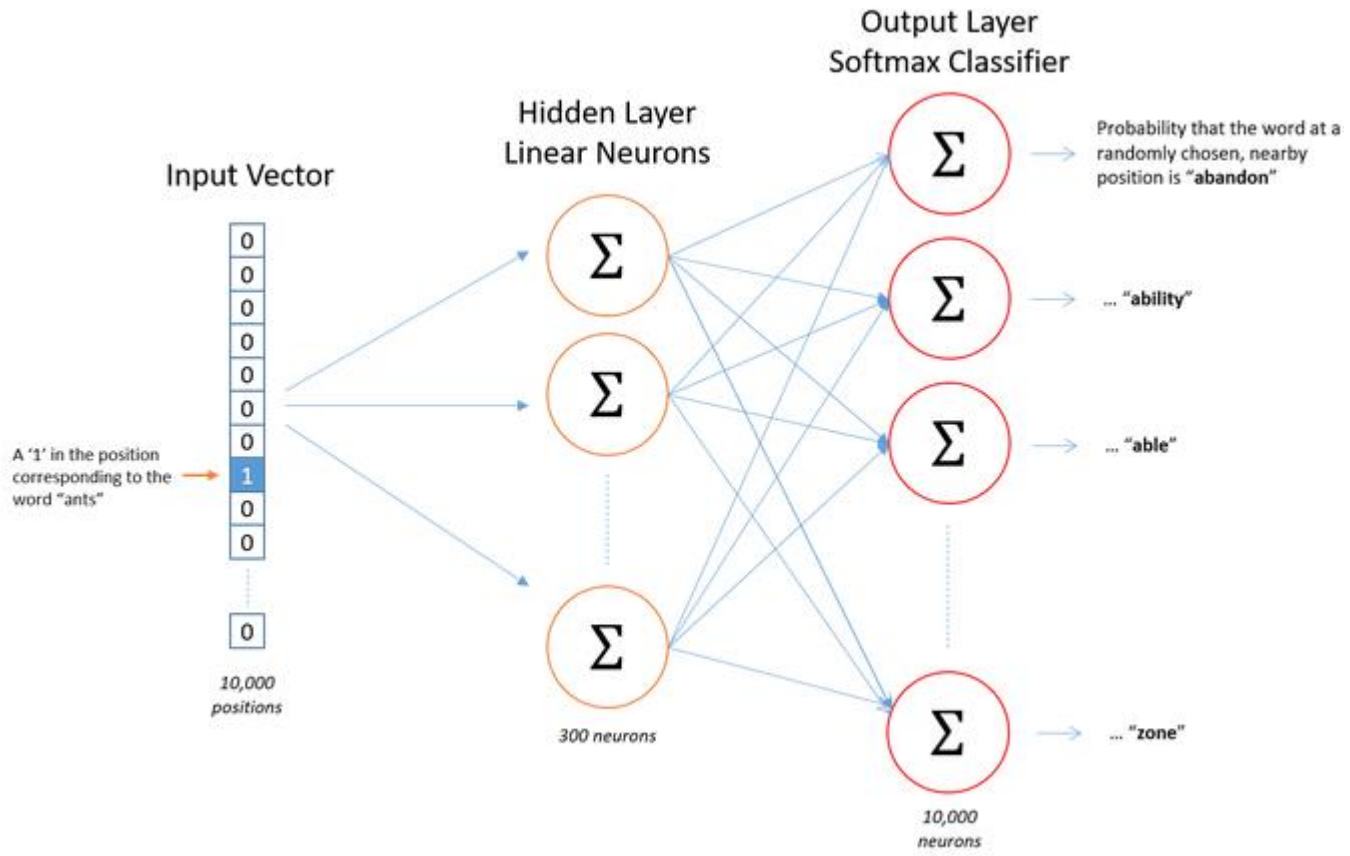
- Word embedding is the collective name for a set of language modeling and feature learning techniques for natural language processing (NLP) where words or sentences from the vocabulary are mapped to vectors of real numbers.

- These vectors are semantically correlated by metrics like cosine distance

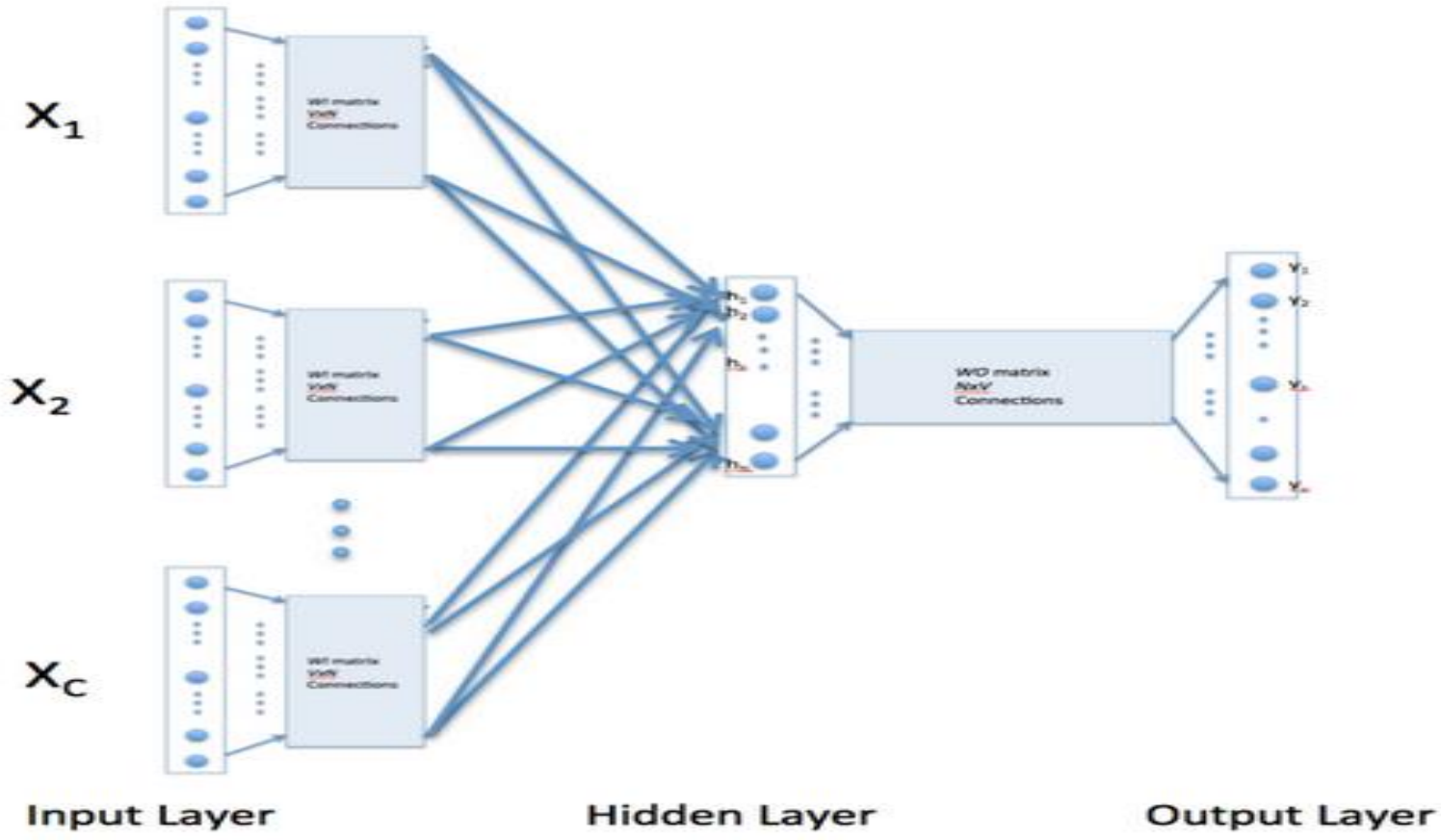



Male-Female | Verb tense | Country-Capital

# Skip-Gram Model (Mikolov, et. al., 2013)



# C-BOW Model (Bow, et al., 2003).

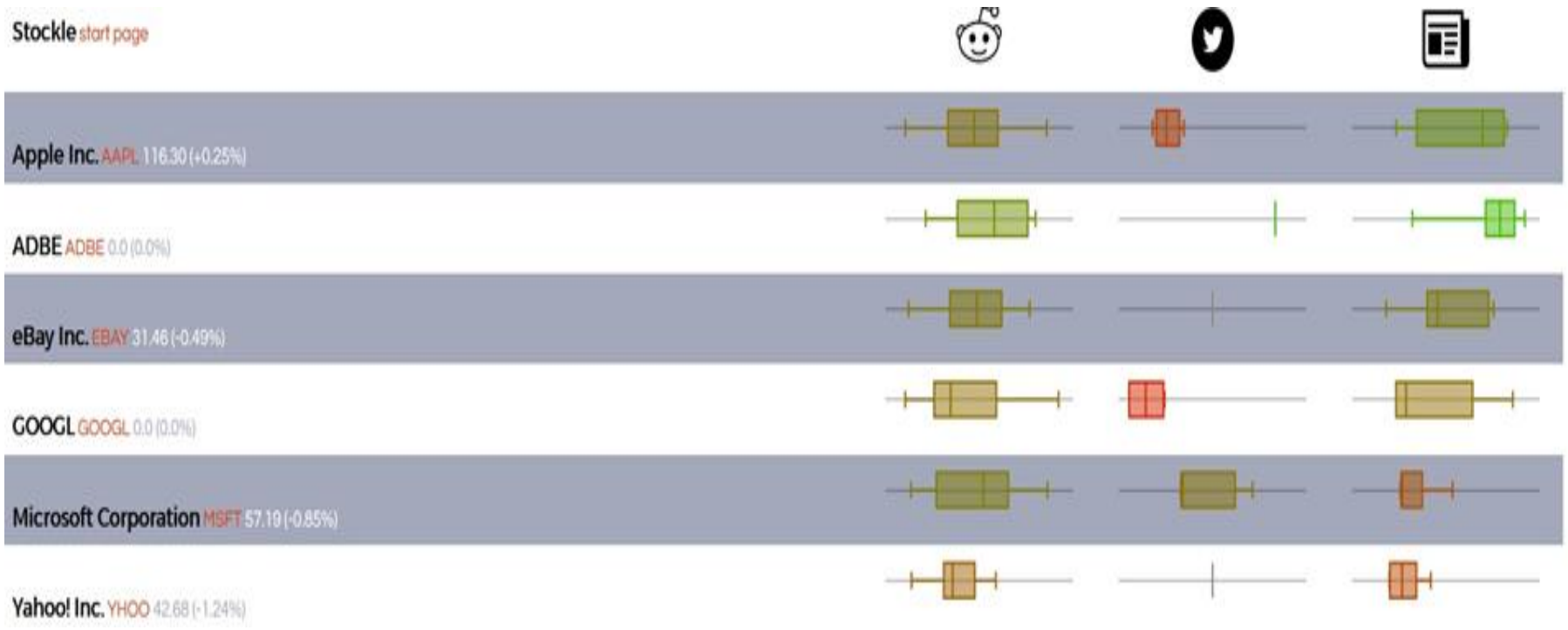


# Sentiment Analysis (*Ain, et al. 2017*)

- **Sentiments** of users that are expressed on the web has great influence on the readers, product vendors and politicians.
- **Sentiment Analysis** refers to text organization for the classification of mind-set or feelings in different manners such as negative, positive, favorable, unfavorable, thumbs up, thumbs down, etc. Thanks to DL, the SA can be visual as well.

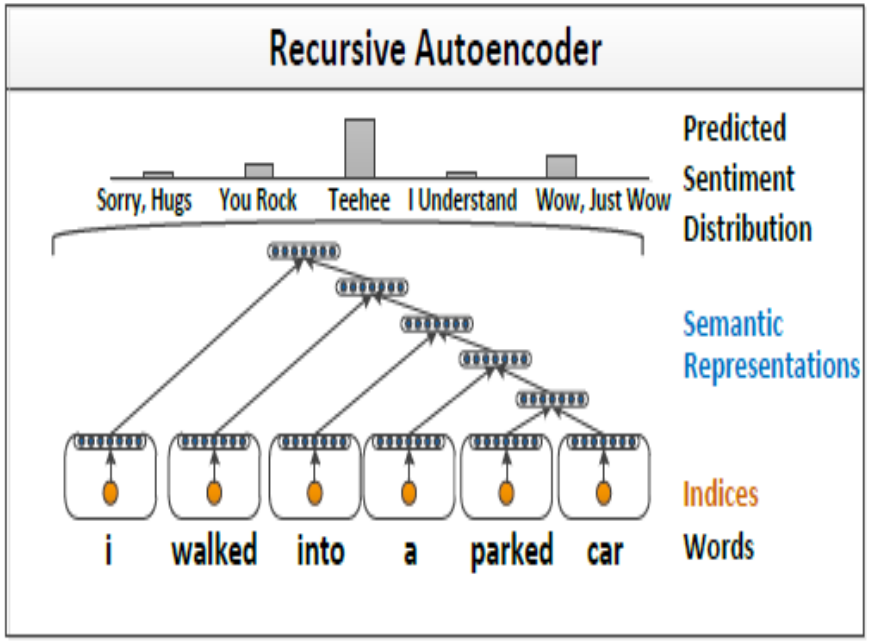
Discovering people opinions, emotions and feelings about a product or service

# Sentiment Analysis with Feedback



# Recursive Neural Tensor Networks (RecursiveNN) (Socher, R., et al., 2011b)

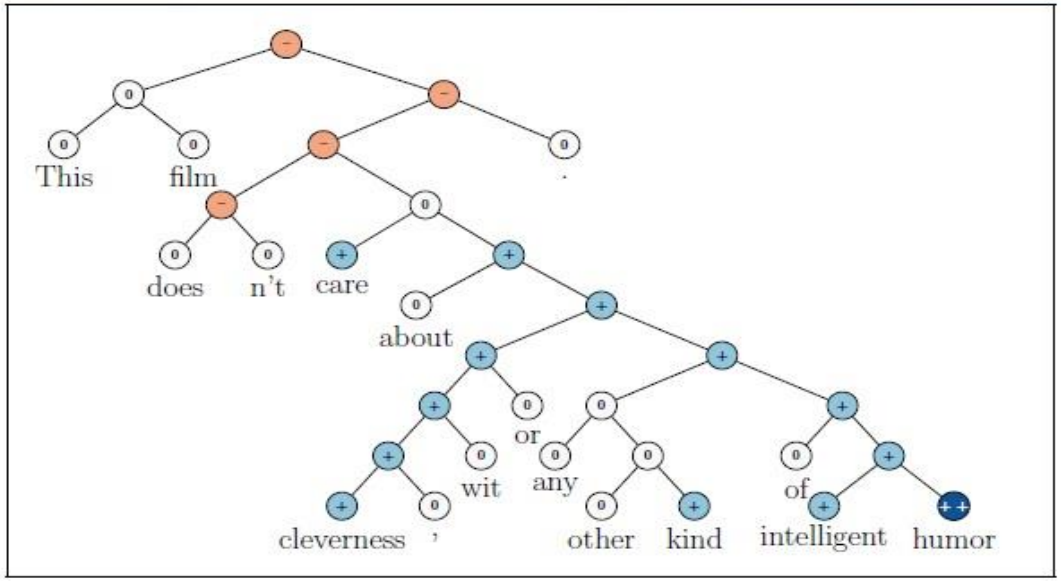
- This models are recursive auto-encoders which learn semantic vector representations of phrases. Word indices (orange) are first mapped into a semantic vector space (blue).
- Then they are recursively merged by the same auto-encoder network into a fixed length sentence representation. The vectors at each node are used as features to predict a distribution over text labels.



# Recursive Neural Tensor Networks (RNTN) (Socher, R., et al. 2013)

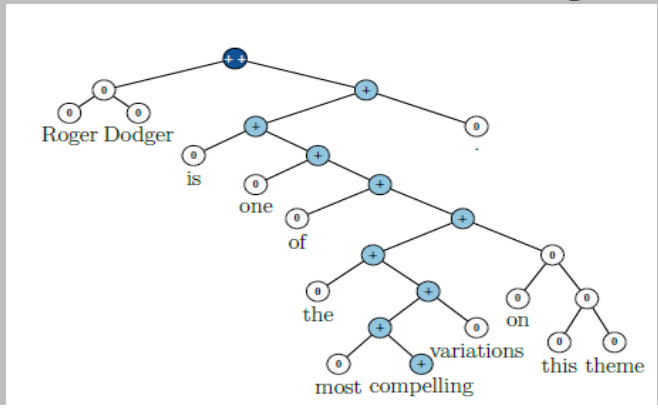
- **The Stanford Sentiment Treebank is the first corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language.**

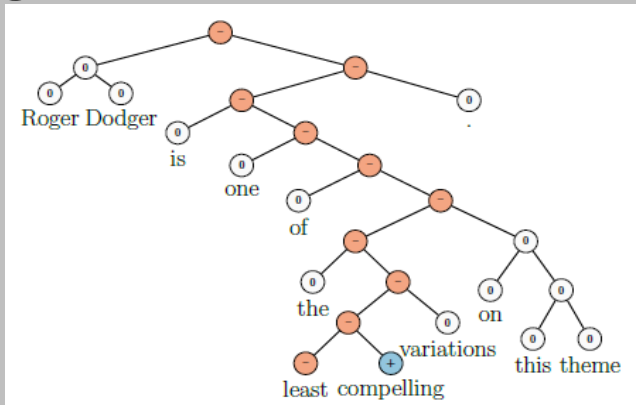
- **RNTNs compute parent vectors in a bottom up fashion using a compositionality function and use node vectors as features for a classifier at that node.**



# RNTN – Upside and Downside

- RNTNS are very efficient in terms of constructing sentence representations.
- RNTNs capture the semantics of a sentence via a tree structure. Its performance heavily depends on the performance of the textual tree construction.
- Constructing such a textual tree exhibits a time complexity of at least O(n2), where n is the length of the text.
- RNTNs are unsuitable for modeling long sentences or documents.



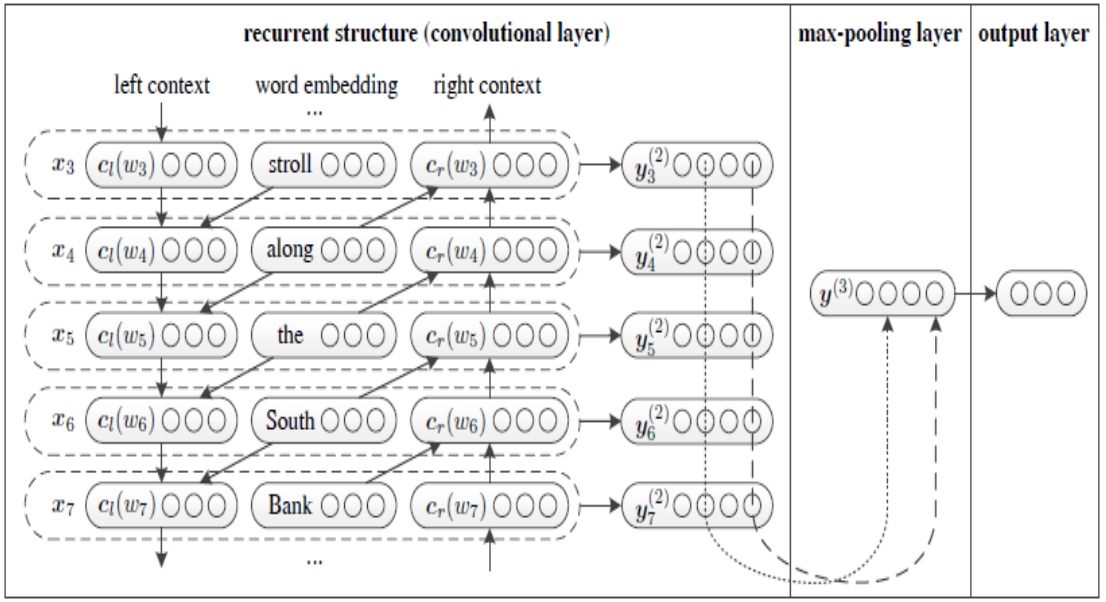



# Recurrent Convolutional Neural Networks (RCNN) (*Lai, S., et al. 2015*)

- They adopt a a recurrent structure to **capture contextual information** as far as possible when learning word representations, which may introduce considerably **less noise compared** to traditional window-based neural networks.

- The **bi-directional recurrent structure** of RCNNs.

- **RCNNs** exhibit a time complexity of O(n)



# RCNN Equations

- RCNNs exhibit a **time complexity of O(n)** , which is linearly correlated with the length of the text length.

- **7 equations** defining all the Neural Network topology

- **Input length** can be variable

$$c_l(w_i) = f(W^{(l)} c_l(w_{i-1}) + W^{(sl)} e(w_{i-1})) \quad (1)$$

$$c_r(w_i) = f(W^{(r)} c_r(w_{i+1}) + W^{(sr)} e(w_{i+1})) \quad (2)$$

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)] \quad (3)$$

$$y_i^{(2)} = \tanh\left(W^{(2)} x_i + b^{(2)}\right) \quad (4)$$

$$y^{(3)} = \max_{i=1}^{n} y_i^{(2)} \quad (5)$$

$$y^{(4)} = W^{(4)} y^{(3)} + b^{(4)} \quad (6)$$

$$p_i = \frac{\exp\left(y_i^{(4)}\right)}{\sum_{k=1}^{n} \exp\left(y_k^{(4)}\right)} \quad (7)$$

# RCNN in Keras

```
class SentimentModelRecConvNet:
    @staticmethod

    def build(input_length, vector_dim):
        hidden_dim_RNN = 200
        hidden_dim_Dense = 100

        embedding = Input(shape=(input_length, vector_dim))

        left_context = LSTM(hidden_dim_RNN, return_sequences = True)(embedding)                                  # Equation 1
        # left_contex: batch_size x tweet_length x hidden_state_dim
        right_context = LSTM(hidden_dim_RNN, return_sequences = True, go_backwards = True)(embedding)            # Equation 2
        # right_cntext: come left_contex
        together = concatenate([left_context, embedding, right_context], axis = 2)                               # Equation 3
        semantic = TimeDistributed(Dense(hidden_dim_Dense, activation = "tanh"))(together)                       # Equation 4
        pool_rnn = Lambda(lambda x: backend.max(x, axis = 1), output_shape = (hidden_dim_Dense, ))(semantic)      # Equation 5
        pool_rnn_args = Lambda(lambda x: backend.argmax(x, axis=1), output_shape = (hidden_dim_Dense, ))(semantic)

        output = Dense(1, input_dim = hidden_dim_Dense, activation = "sigmoid")(pool_rnn)                        # Equations 6, 7

        deepnetwork = Model(inputs=embedding, outputs=output)
        deepnetwork_keywords = Model(inputs=embedding, outputs=pool_rnn_args)

        return [deepnetwork, deepnetwork_keywords]
```

# RCNN: Feature Extraction

- RCNNs employ a max-pooling layer that automatically judges which words play key roles in text classification to capture the key components in texts.

- The most important words are the information most frequently selected in the max-pooling layer.

- Contrary to the most positive and most negative phrases in RNTN, RCNN does not rely on a syntactic parser, therefore, the presented n-grams are not typically “phrases”.

| | RCNN |
|---|---|
| P | well *worth* the; a *wonderful* movie; even *stinging* at; and *invigorating* film; and *ingenious* entertainment; and *enjoy* .; ’s *sweetest* movie |
| N | A *dreadful* live-action; Extremely *boring* .; is *n’t* a; ’s *painful* .; Extremely *dumb* .; an *awfully* derivative; ’s *weaker* than; incredibly *dull* .; very *bad* sign; |
| | **RNTN** |
| P | an amazing performance; most visually stunning; wonderful all-ages triumph; a wonderful movie |
| N | for worst movie; A lousy movie; a complete failure; most painfully marginal; very bad sign |

# RCNN applied to Extractive Text Summarization

- **Best keywords** lead to **best contextes** ---> **Summarization**

```
Tweet 29: "GiÃ  avete letto 136 pagine del piano scuola? #Fenomeni #labuonascuola"

Sentiment: -0.95 - -1
Keywords: pagine, avete, fenomeni, piano

Tweet 30: "\"Per l'#aternanza #scuola #lavoro bisogna passare da 11a 100milioni di euro\" #labuonascuola http://t.co/zGAzkni8rv"

Sentiment: -0.81 - -1
Keywords: euro, t, scuola, lavoro




Most significant keywords driving the sentiment decision:

Eccolo

Siamo

Scuola

Giuste

Escluso


Most significant sentences driving the sentiment decision:

...cambierÃ  solo se noi metteremo al centro...

...solo se noi metteremo al centro la...

...piÃ¹ grande spettacolo mai visto passodopopasso scuola...

...mai visto passodopopasso scuola labuonascuola...

...nessuno si senta escluso la buona scuola...
```

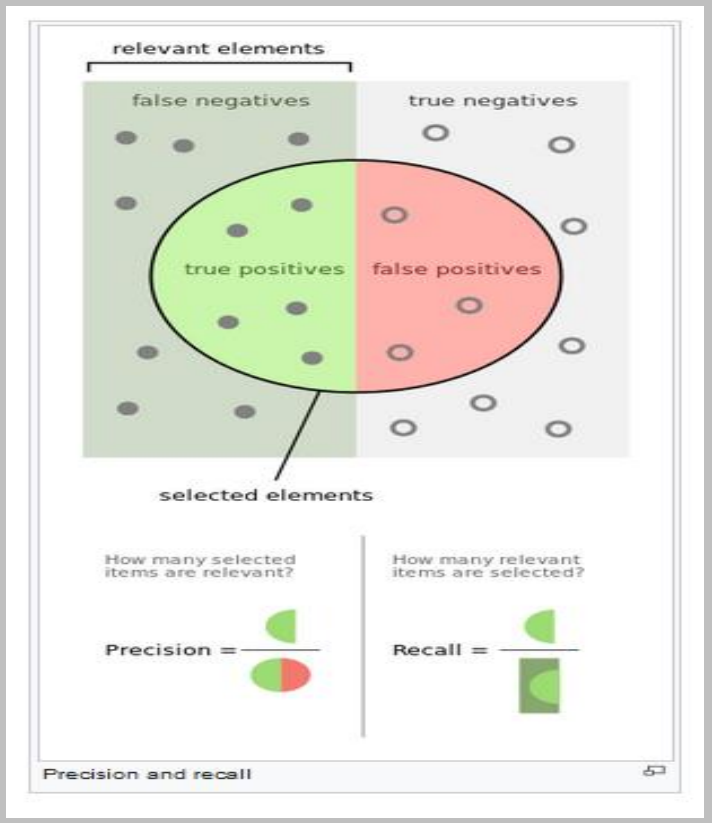
# Recurrent Neural Networks are able to understand negations and other things

- Thanks to **word embeddings** semantics **RNNs** can recognize **nagations**, and complex **forms** of **language utterances**.

```
Tweet: This is a bad thing
- Sentiment: -0.72 - -1

Keywords: bad, thing, a, is

Tweet: This is not a bad thing
- Sentiment: 0.46 - +1

Keywords: not, thing, bad, a

Tweet: This is a positive thing
- Sentiment: 0.94 - +1

Keywords: positive, thing, a, is

Tweet: This is a very positive thing
- Sentiment: 0.91 - +1

Keywords: positive, very, thing, a

Tweet: I like Renzi politics
- Sentiment: 0.70 - +1

Keywords: like, renzi, politics, i

Tweet: I don't agree with Renzi Politics
- Sentiment: 0.16 - 0

Keywords: don't, agree, politics, renzi

Tweet: Renzi did a wrong international Politics
- Sentiment: -0.34 - -1

Keywords: wrong, did, renzi, international

Tweet: Renzi did a very good international Politics
- Sentiment: 0.74 - +1

Keywords: did, renzi, good, very

Tweet: Istat is a very good Institute of research
- Sentiment: 0.84 - +1

Keywords: good, very, research, istat

Tweet: Istat is not a good Institute of research - Sentiment: -0.78 - -1

Keywords: not, research, istat, institute
```

# Classification Metrics

## F-score



Precision and recall

**sensitivity, recall, hit rate, or true positive rate (TPR)**

$$\mathrm{TPR} = \frac{\mathrm{TP}}{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

**specificity or true negative rate (TNR)**

$$\mathrm{TNR} = \frac{\mathrm{TN}}{N} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

**precision or positive predictive value (PPV)**

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

**negative predictive value (NPV)**

$$\mathrm{NPV} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}}$$

**miss rate or false negative rate (FNR)**

$$\mathrm{FNR} = \frac{\mathrm{FN}}{P} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} = 1 - \mathrm{TPR}$$

**fall-out or false positive rate (FPR)**

$$\mathrm{FPR} = \frac{\mathrm{FP}}{N} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} = 1 - \mathrm{TNR}$$

**false discovery rate (FDR)**

$$\mathrm{FDR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TP}} = 1 - \mathrm{PPV}$$

**false omission rate (FOR)**

$$\mathrm{FOR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TN}} = 1 - \mathrm{NPV}$$

**accuracy (ACC)**

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

**F1 score**

is the harmonic mean of precision and sensitivity

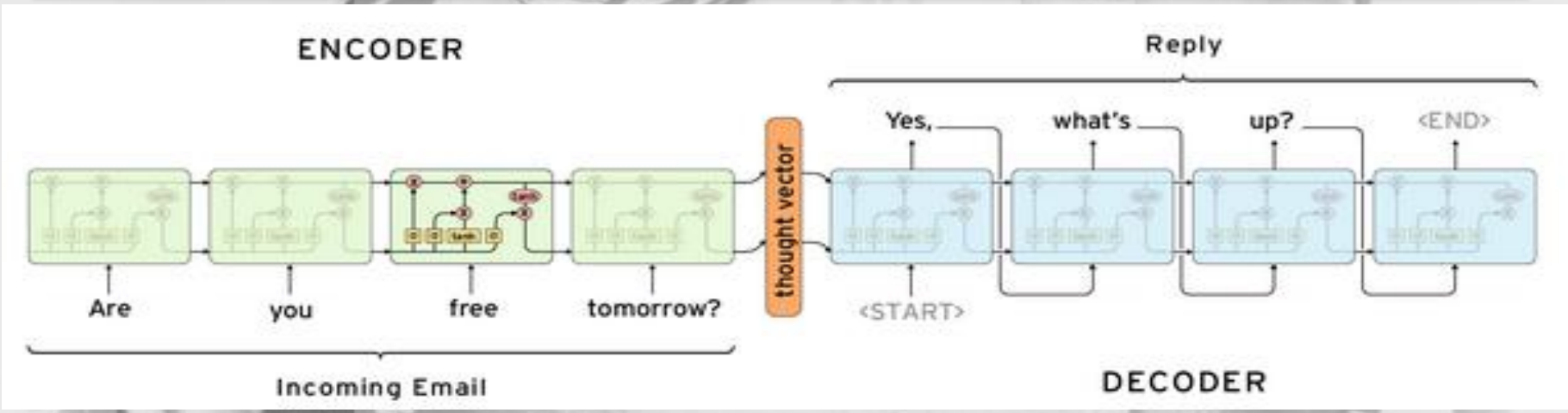
$$F_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

# Use Case 2: Classification of Cifar 10 with CNN in Keras

## Metrics

| | | True condition | | | |
|---|---|---|---|---|---|
| | Total population | Condition positive | Condition negative | Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$ | Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$ |
| **Predicted condition** | Predicted condition positive | **True positive**, Power | **False positive**, Type I error | Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$ | False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$ |
| | Predicted condition negative | **False negative**, Type II error | **True negative** | False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$ | Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$ |
| | | True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$ | False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$ | Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$ | Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$ |
| | | False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$ | True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$ | Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$ | $F_1$ score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$ |

# Neural Conversational Models (Vinyals, & Le., 2015).




## Conversation model – chatbot?

- Training on a set of conversations. The input sequence can be the concatenation of what has been conversed so far (the context), and the output sequence is the reply.

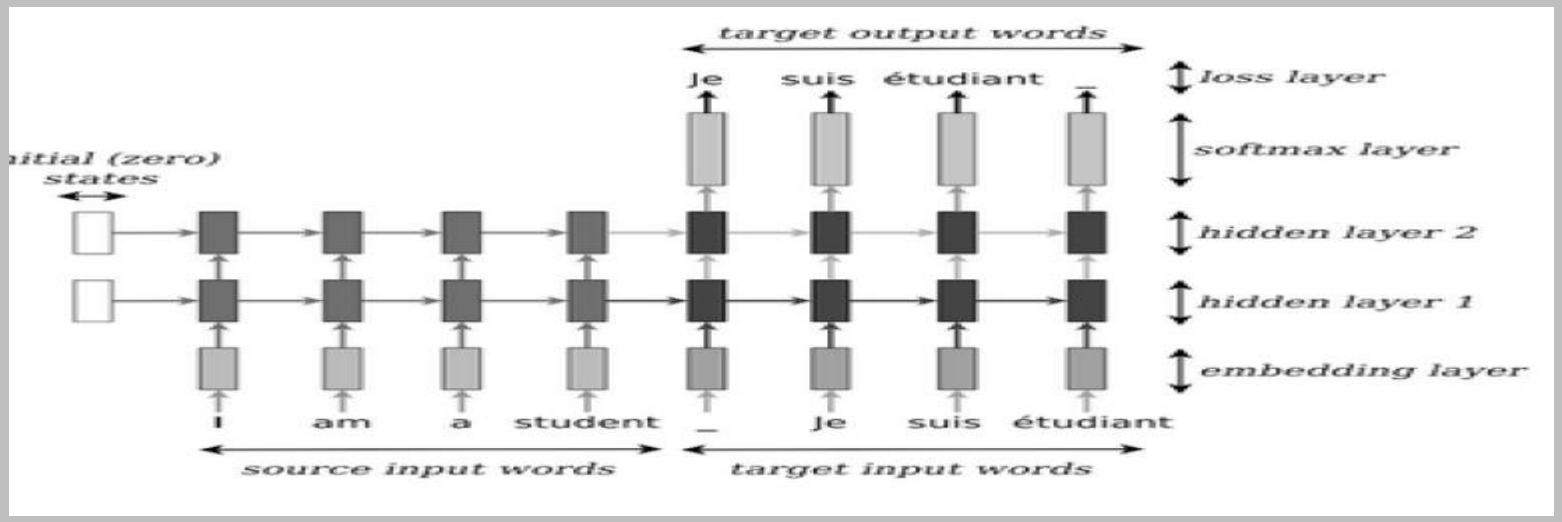
Conversation 4: Philosophical Q&A

**Machine:** *is it a windows machine ?*
**Machine:** *we have linux on your machine .*
**Machine:** *is it windows xp or windows 7 ?*
**Human:** *i am using linux*
**Machine:** *let me take a remote of your machine and check if issue has been pushed properly .*
**Human:** *ok*
**Machine:** *please open terminal and run the below commands in one computer :*
**Machine:** *sudo apt - get update*
**Machine:** *sudo apt - get upgrade*
**Machine:** *sudo <COMMAND> - updater*
**Human:** *done*
**Machine:** *let me try connecting now*
**Human:** *ok*

**Human:** *what is the purpose of life ?*
**Machine:** *to serve the greater good .*
**Human:** *what is the purpose of living ?*
**Machine:** *to live forever .*
**Human:** *what is the purpose of existence ?*
**Machine:** *to find out what happens when we get to the planet earth .*
**Human:** *where are you now ?*
**Machine:** *i 'm in the middle of nowhere .*
**Human:** *what is the purpose of dying ?*
**Machine:** *to have a life .*
**Human:** *what is the purpose of being intelligent ?*
**Machine:** *to find out what it is .*
**Human:** *what is the purpose of emotions ?*
**Machine:** *i don 't know .*

# Neural Machine Translation (Bahdanau, et., al., 2014).

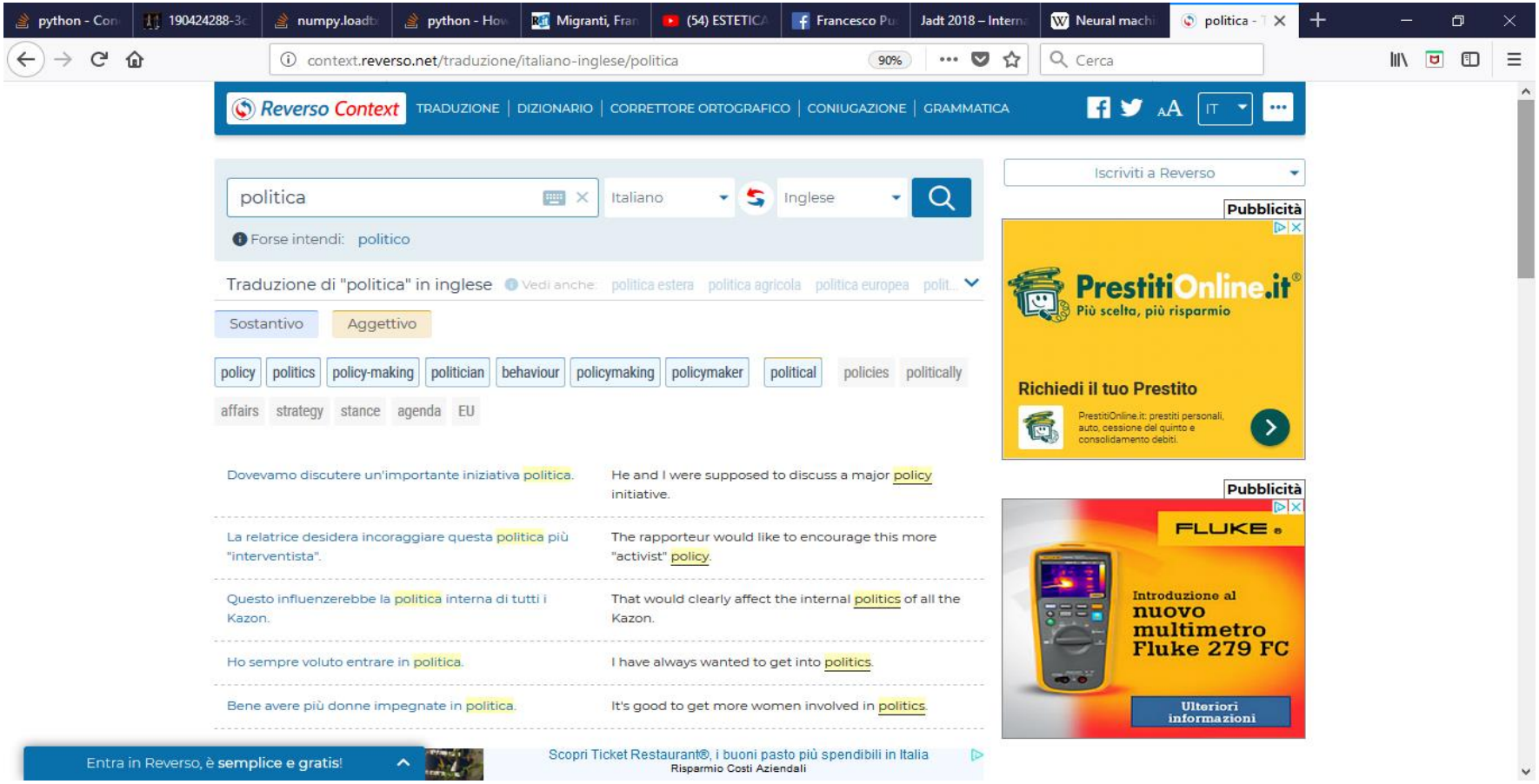
- **Neural machine translation (NMT) is an approach to "machine translation" that uses large ANN to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model (Bahdanau et al., 2014; Luong et Manning, 2016). .**



# Multilingual Sentiment Analysis

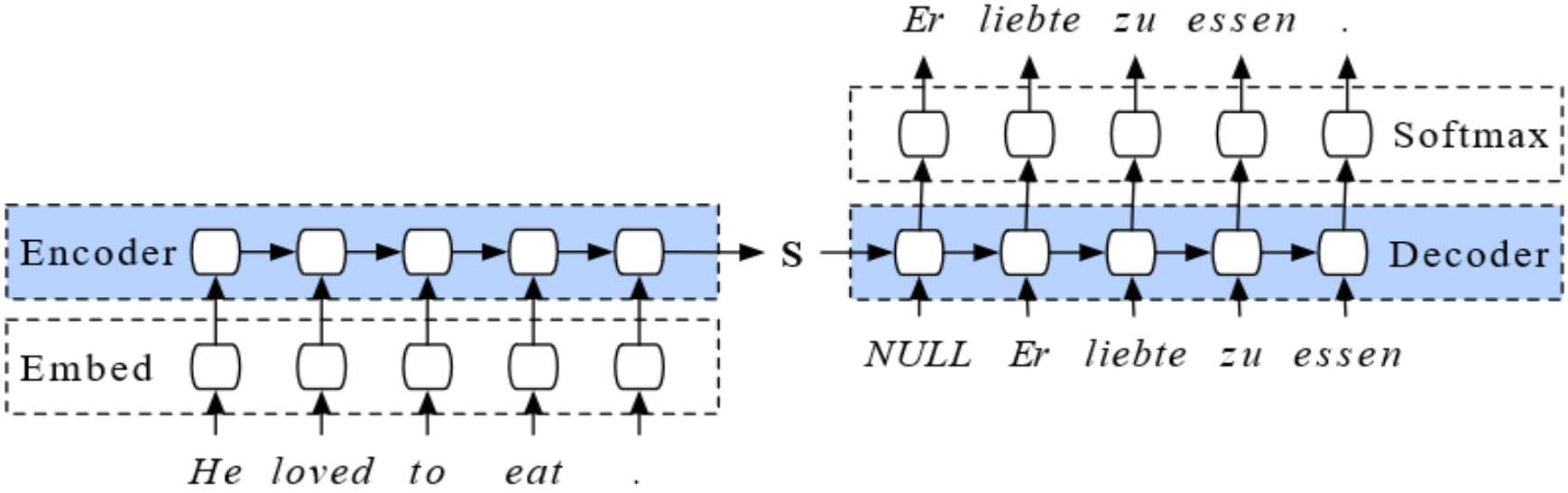
- During the training stage, the RCNN achieves 84% of accuracy on a validation set (selected at the 20% of the original dataset). On a test set of 380 tweets (provided by Semeval), the model returns around 82% of accuracy on positive tweets and 78% of accuracy on negatives, with an approximative 80% overall on a mixed tweets set.
- During the training we determined 3.2 millions of keywords, namely 2 for each tweet, the most important and the second in order of signinificancy.

# Contextual Translations Web-sites



# Neural Machine Translation

| *Input sentence:* | *Translation (PBMT):* | *Translation (GNMT):* | *Translation (human):* |
|---|---|---|---|
| 李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。 | Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session. | Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers. | Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada. |



# Neural Machine Translation

adottare un vocabolario condiviso è un suggerimento perfetto su come scrivere frasi comprensibili
adopt a shared vocabulary is a perfect suggestion on how to write understandable sentences

un altro suggerimento su come scrivere frasi semplici: evita le negazioni inutili
another suggestion about how to write simple sentences : avoid unnecessary <unk>

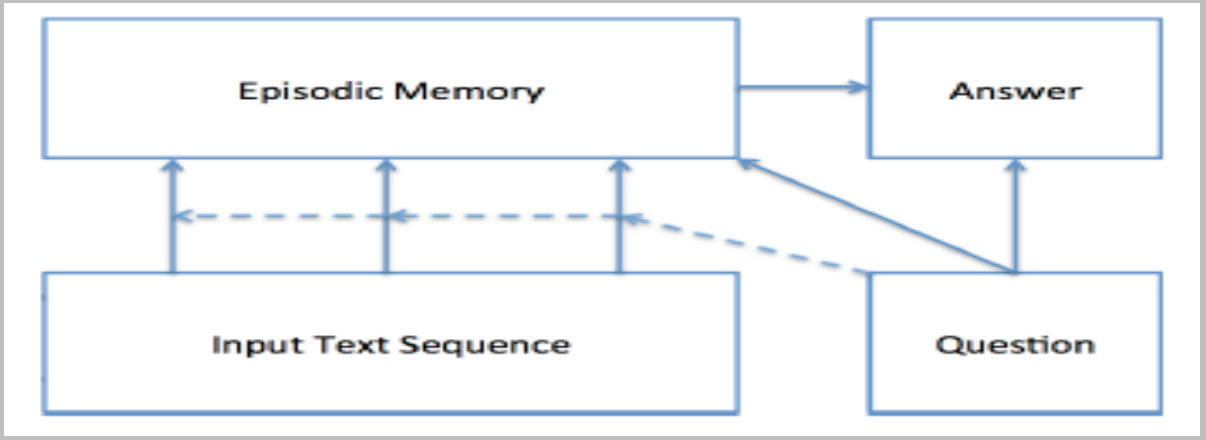
quasi 90 persone sono morte per una tempesta tropicale nelle filippine
nearly 90 people died for a tropical storm in the philippines

*Figure 3. Some translations from Italian to English by means of the neural model trained by us.*

- We have tested the English RCNN model on the same italian SENTIPOLC 2016 test-set translated into English by our neural machine translation model. Results highlight a boost of performance : **78%** of accuracy on the test set versus the **43%** of the Italian trained RCNN model proving our strategy of stacking NMT and RCNN models is successful.

# Dynamic Memory Networks (Kumar, et al., 2016).

- Dynamic Memory Networks (DMN) are a recurrent neural network architecture which processes input sequences and questions, forms episodic memories, and generates relevant answers. The DMN can be trained end-to-end and obtains state-of-the-art results on question answering (Facebook's bAbI dataset), text classification for sentiment analysis (Stanford Sentiment Treebank) and sequence modeling for part-of-speech tagging (WSJ-PTB).



# Dynamic Memory Networks (DMN)

I: Jane went to the hallway.
I: Mary walked to the bathroom.
I: Sandra went to the garden.
I: Daniel went back to the garden.
I: Sandra took the milk there.
Q: Where is the milk?
A: garden
I: It started boring, but then it got interesting.
Q: What's the sentiment?
A: positive
Q: POS tags?
A: PRP VBD JJ , CC RB PRP VBD JJ .

**Task 1: Single Supporting Fact**
Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

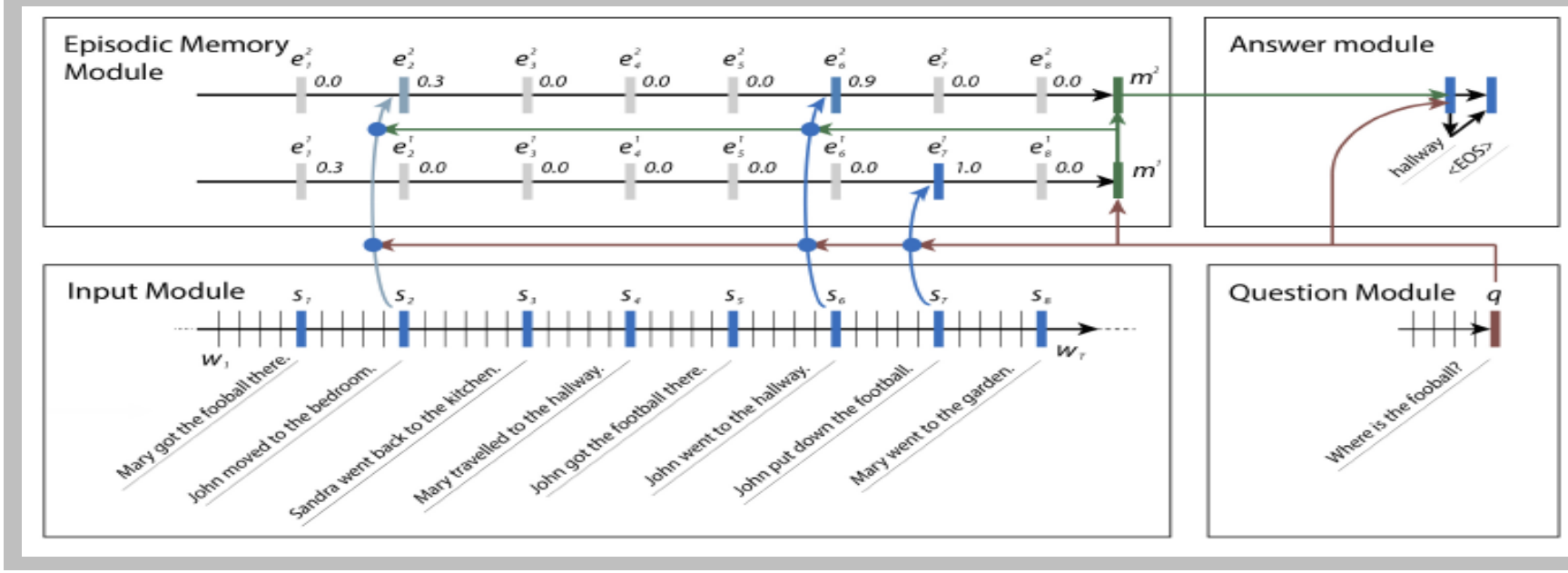
**Task 2: Two Supporting Facts**
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

**Task 3: Three Supporting Facts**
John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

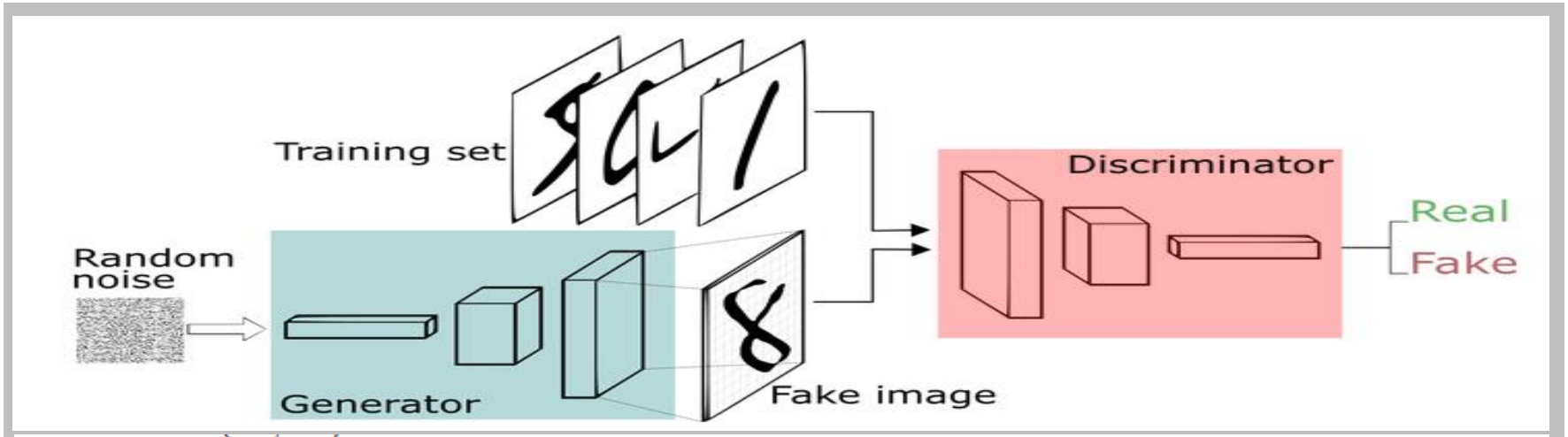
**Task 4: Two Argument Relations**
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

**Task 5: Three Argument Relations**
Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

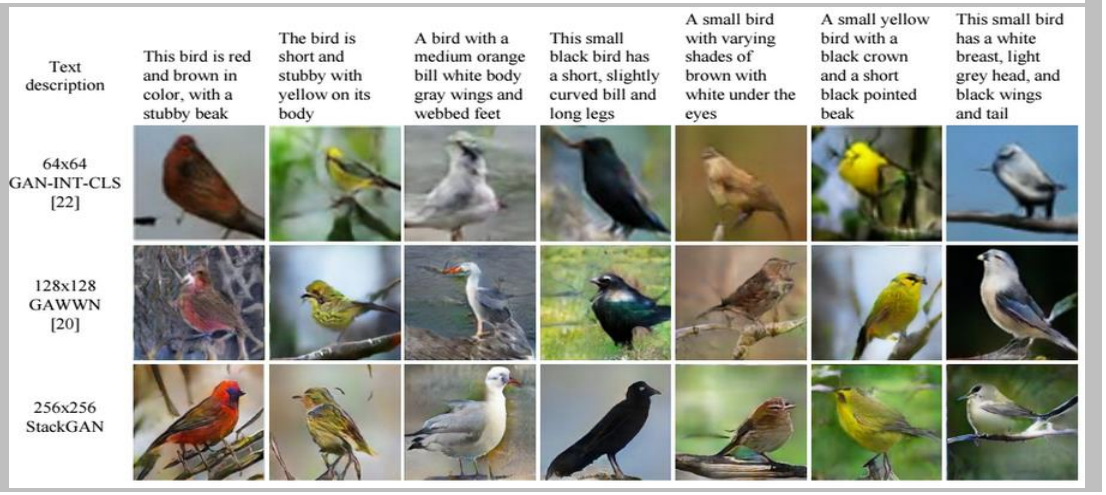
**Task 6: Yes/No Questions**
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes



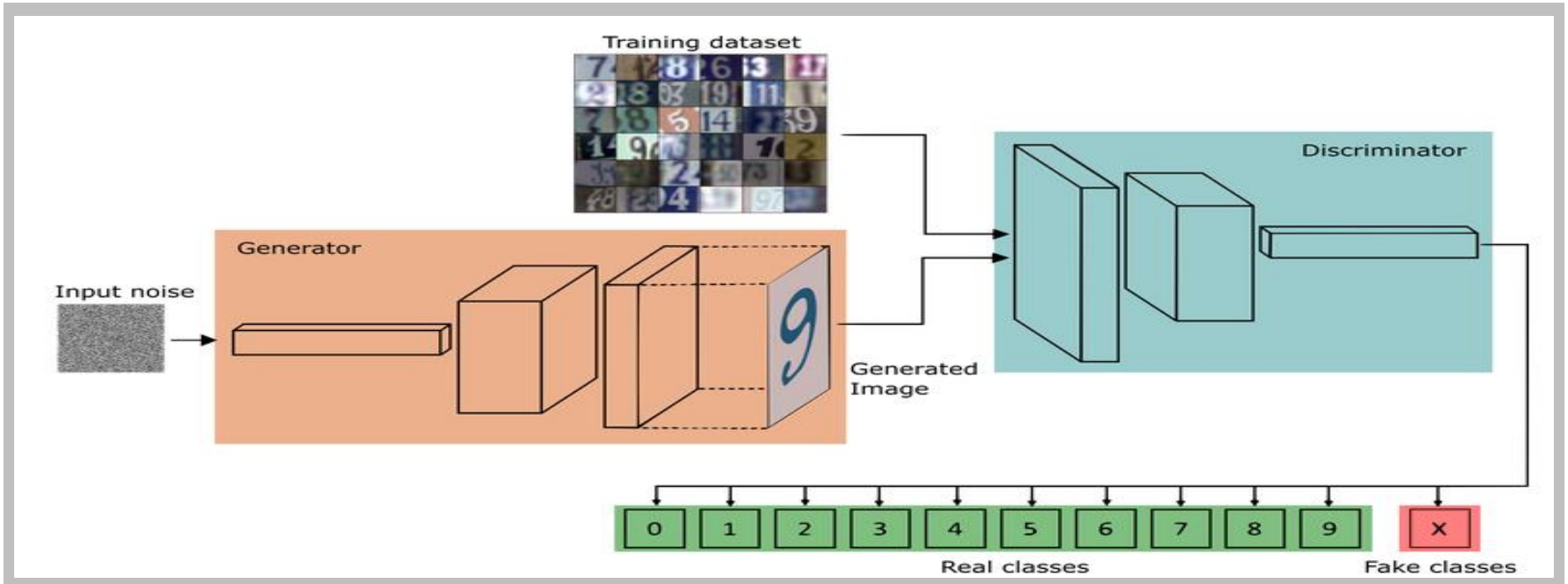
# Generative Adversarial Networks (GAN) (Goodfellow, et al.,2014)




$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$



# Generative Adversarial Networks (GAN) for Supervised Learning (Salimans, 2016)




$$L = -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y = K+1|\boldsymbol{x})]$$

$$= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where}$$

$$L_{\text{supervised}} = -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)} \log p_{\text{model}}(y|\boldsymbol{x}, y < K+1)$$

$$L_{\text{unsupervised}} = -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log[1 - p_{\text{model}}(y = K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G} \log[p_{\text{model}}(y = K+1|\boldsymbol{x})]\}$$

$$L_{\text{unsupervised}} = -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log D(\boldsymbol{x}) + \mathbb{E}_{z\sim \text{noise}} \log(1 - D(G(\boldsymbol{z})))\}.$$

# USEFUL Links


- **EMOS PROJECT – Prof. Agostino di Ciaccio**

**WebSite:**
**http://ec.europa.eu/eurostat/web/european-statistical-system/emos**

# REFERENCES


**Scott, A. J., & Knott, M. (1974).** A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 507-512.

**Vinyals, O., & Le, Q. (2015).** A neural conversational model. *arXiv preprint arXiv:1506.05869*.

**Bahdanau, D., Cho, K., & Bengio, Y. (2014).** Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

**Sutskever I, Hinton GE (2008)**. Deep, narrow sigmoid belief networks are universal approximators. Neural Computation. 20(11):2629–36. pmid:18533819

**Roux NL, Bengio Y. (2010)** Deep Belief Networks Are Compact Universal Approximators. Neural Computation.22(8):2192–207.

**Amari, S. I., Cichocki, A., & Yang, H. H. (1995, December).** Recurrent neural networks for blind separation of sources. In *Proc. Int. Symp. NOLTA* (pp. 37-42).

**Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014).** Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

**Hochreiter S, Schmidhuber J. Long Short-Term Memory (1997)**. Neural Computation. 1997;9(8):1735–80. pmid:9377276

# REFERENCES


**Bliemel F. Theil's** (1973) Forecast Accuracy Coefficient: A Clarification. Journal of Marketing Research. 10(4):444.

**Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., ... & Socher, R.** (2016, June). Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning* (pp. 1378-1387).

**Bow, C., Hughes, B., & Bird, S. (2003, July).** Towards a general model of interlinear text. In *Proceedings of EMELD workshop* (pp. 11-13).

**Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013).** Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

**Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014).** Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

**Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016).** Improved techniques for training gans. In *Advances in Neural Information Processing Systems* (pp. 2234-2242).

# AKNOWLEDGEMENTS

THANK YOU

FOR YOUR ATTENTION

Francesco Pugliese