Natural Language Processing applied to Finance

Grégory Bournassenko

Université de Paris
Department of law, economics and management

October 2021

1. What is NLP
2. Use cases in finance
3. Exemple of application
4. Drawbacks
5. Bibliography

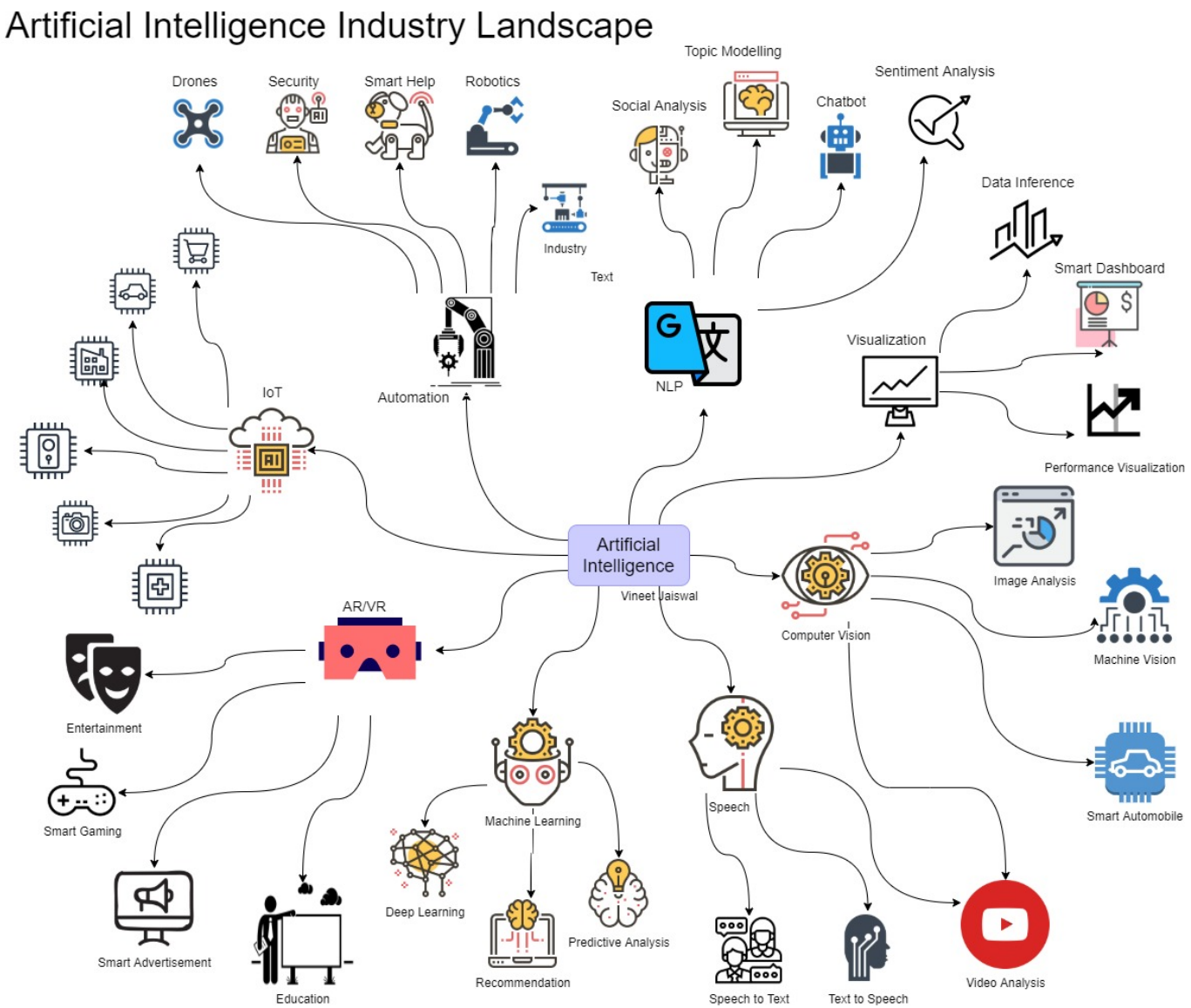
NLP ?

Definition

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

Goal

The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.


Artificial Intelligence Industry Landscape
Drones
Security
Smart Help
Robotics
Industry
Topic Modelling
Social Analysis
Chatbot
Sentiment Analysis
Data Inference
Smart Dashboard
Visualization
Performance Visualization
Text
NLP
Automation
IoT
Artificial Intelligence
Vineet Jaiswal
Image Analysis
Computer Vision
Machine Vision
AR/VR
Entertainment
Smart Gaming
Smart Advertisement
Education
Machine Learning
Deep Learning
Recommendation
Predictive Analysis
Speech
Smart Automobile
Speech to Text
Text to Speech
Video Analysis

Example : Sentiment Analysis

Classic data processing

String a = "I buy TSLA";
String b = "I sell TSLA";
a.equals(b);
$\gg$ FALSE
Because b $\neq$ s, u $\neq$ e ...

Natural Language Processing

String a = "I buy TSLA";
String b = "I sell TSLA";
findSentiment(a).equals(findSentiment(b));
$\gg$ FALSE
Because buy $\neq$ sell in the meaning

How it works

3 algorithms :

How it works

3 algorithms :

Rule-based : sentiment analysis based on a set of manually crafted rules.

How it works

3 algorithms :

Rule-based : sentiment analysis based on a set of manually crafted rules.

Automatic : machine learning techniques to learn from data.

How it works

3 algorithms :

Rule-based : sentiment analysis based on a set of manually crafted rules.

Automatic : machine learning techniques to learn from data.

Hybrid : both rule-based and automatic approaches.

Risk assessments (loans...)

Risk assessments (loans...)

Financial sentiment analysis (positive $\neq$ positive market reaction), BioBERT, FinBERT...

Risk assessments (loans...)

Financial sentiment analysis (positive $\neq$ positive market reaction), BioBERT, FinBERT...

Accounting and auditing (document reviews...)

Risk assessments (loans...)

Financial sentiment analysis (positive $\neq$ positive market reaction), BioBERT, FinBERT...

Accounting and auditing (document reviews...)

Portfolio selection & optimization (filter desirable & undesirable stocks)

Just an example

Algorithm idea :

1) We define an asset universe U

2) We get U_i's underlying sentiment S_t at a time t

3) Then we get S_{t+x} at a time $_{t+x}$ where $x > 0$, a fixed delay.

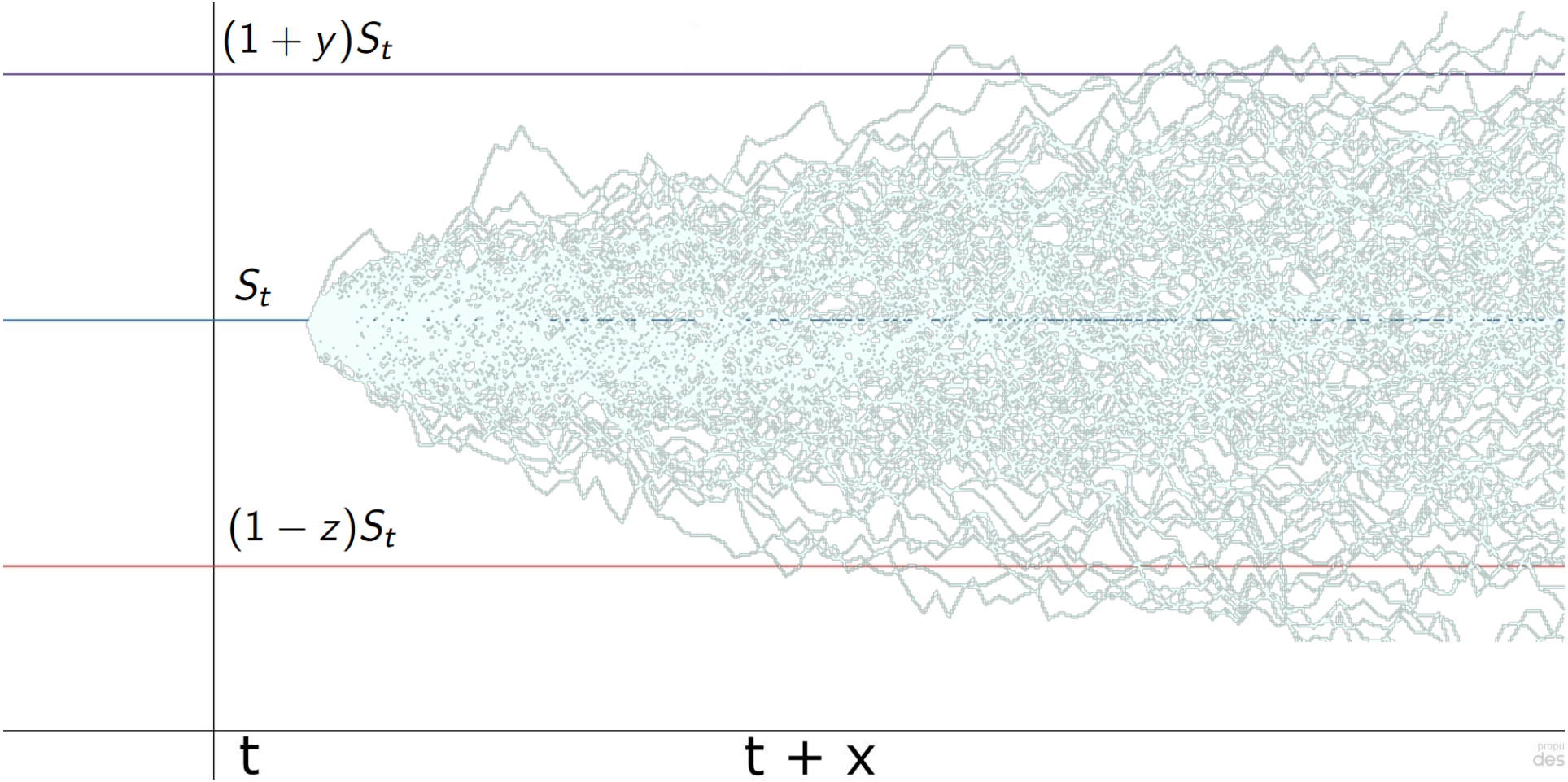
4) If $S_{t+x} > (1 + y)S_t$, we open a long position for U_i, where $0 \leq y \leq 1$ representing the equivalent of a "take profit".

5) Else if $S_{t+x} < (1 - z)S_t$, we open a short position for U_i, where $0 \leq z \leq 1$ representing the equivalent of a "stop loss".

Just an example

6) Else if $(1-z)S_t < S_{t+x} < (1+y)S_t$, we wait a fixed time t and reiterate from step 2.

We can go much further in the algorithm:
- what to do after a long position opened ?
- what to do after a short position ?
- best values for x, y, z ?
- how many confirmations (out of boundaries) before positions opening / closing ?


$(1+y)S_t$
S_t
$(1-z)S_t$
t
t + x

Just an example

```
def get_twitter_BTC():
    ratios = 0
    for keyword in keywords:
        tweets = tweepy.Cursor(api2.search, q=keyword, lang="en").items(nb)
        pos = 0
        neg = 0
        for tweet in tweets:
            analysis = TextBlob(tweet.text)
            if 0 <= analysis.sentiment.polarity <= 1:
            pos += 1
            elif -1 <= analysis.sentiment.polarity < 0:
            neg += 1
        pos = perc(pos, nb)
        neg = perc(neg, nb)
        if float(neg) > 0:
            ratio = float(pos) / float(neg)
        else:
            ratio = float(pos)
        ratios += ratio
    return ratios
```

Just an example

```
for k in range(1000):
    score = get_twitter_BTC()
    min1 = score + (score * 30 / 100)
    time.sleep(60*5)
    new_score = get_twitter_BTC()

    if new_score > min1:
        btc_price = get_current_price("BTC-USD")
        buy = "\nBUY : " + str(btc_price)
        f = open("output.txt", "a")
        f.write(buy)
        f.close()
        time.sleep(60*5)
        new_new_score = get_twitter_BTC()
        min2 = new_score - (new_score * 30 / 100)

        if new_new_score < min2:
            new_btc_price = get_current_price("BTC-USD")
            sell_at = " SELL : + str(new_btc_price)
            trade_profit = new_btc_price - btc_price
            perc_profit = trade_profit / btc_price * 100
            perc_profit_round = round(perc_profit, 3)
            sell_message = sell_at + " | " + " Profit = " + str(perc_profit_round) + " %" f = open("output.txt", "a")
            f.write(sell_message)
            f.close()
            time.sleep(60*5)
    else:
        time.sleep(60*5)
```

Context

Context

"What do you like about TSLA ? : Everything !"

Context

"What do you like about TSLA ? : Everything !"
"What do you hate about TSLA ? : Everything !"

Context

"What do you like about TSLA ? : Everything !"
"What do you hate about TSLA ? : Everything !"

Irony & sarcasm

Context

"What do you like about TSLA ? : Everything !"
"What do you hate about TSLA ? : Everything !"

Irony & sarcasm

"BTC is not volatile enough"

Context

"What do you like about TSLA ? : Everything !"
"What do you hate about TSLA ? : Everything !"

Irony & sarcasm

"BTC is not volatile enough"

Information weight

Elon Musk : "Let's buy TSLA"

Context

"What do you like about TSLA ? : Everything !"
"What do you hate about TSLA ? : Everything !"

Irony & sarcasm

"BTC is not volatile enough"

Information weight

Elon Musk : "Let's buy TSLA"
Someone else : "Let's buy TSLA"

Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L. T., & Trajanov, D. (2020). Evaluation of sentiment analysis in finance: from lexicons to transformers. IEEE Access, 8, 131662-131682.

Costantino, M., Collingham, R. J., & Morgan, R. G. (1996). Qualitative Information in Finance: Natural Language Processing and Information Extraction. NeuroVe $ t Journal, 4(6).

“5 Natural Language Processing (NLP) Applications In Finance.” Avenga, 3 Feb. 2021, https://www.avenga.com/magazine/nlp-finance-applications/.

"6 Real-World Examples of Natural Language Processing." Expert.Ai, 28 June 2021, https://www.expert.ai/blog/natural-language-processing-examples/.

"Four Ways to Apply NLP in Financial Services." Refinitiv Perspectives, 28 Apr. 2020, https://www.refinitiv.com/perspectives/ai-digitalization/four-ways-to-apply-nlp-in-financial-services/.

internationalbanker. "How the Financial Industry Is Using Natural Language Processing." International Banker, 12 Apr. 2021, http://internationalbanker.com/technology/how-the-financial-industry-is-using-natural-language-processing/.

"Sentiment Analysis: The Go-To Guide." MonkeyLearn, https://monkeylearn.com/sentiment-analysis/. Accessed 4 Oct. 2021.

"Step By Step Guide To Natural Language Processing (NLP) In Trading." Quantitative Finance & Algo Trading Blog by QuantInsti, 22 July 2019, https://blog.quantinsti.com/natural-language-processing-trading/.

"These Popular NLP Applications Are Changing the Face of Finance." Day One: AI Development Services , App Development Company, 1 June 2021, https://www.day1tech.com/these-popular-nlp-applications-are-changing-the-face-of-finance/.

"Why Finance Is Deploying Natural Language Processing." MIT Sloan, https://mitsloan.mit.edu/ideas-made-to-matter/why-finance-deploying-natural-language-processing. Accessed 4 Oct. 2021.

Zimmerman, Vered. "How the Large Investment Firms Use NLP." Medium, 21 May 2021, https://towardsdatascience.com/how-the-large-investment-firms-use-nlp-822c7c79af96.