



atirso

Machine Learning for text Analytics

 **idata**[®]
Advanced analytics for business



Unstructured information increases

exponentially in the world.





Challenges

Process large volumes of unstructured data in a short time.

Give value to the thoughts and opinions of your customers or employees..

The information resolved day by day can be impossible to analyze for or several people in short times, but not for a robot.



Ideal Solution

Classification of your contratio, documents automatically using Machine learning and textmining tools.

Real time analysis of the information.

Agility in the processes of analysis and classification of unstructured data, which means lower costs and more efficiency.



Desired Outcomes

A restructuration of the information capture and analysis model.

Robot that allows continuous learning, with which the classification model would be constantly updated.

TIRSO can provide entrepreneurs with the necessary tools to allocate resources or propose strategies to keep the company at the forefront of the market.

A robotic hand is shown holding a tablet computer. The hand is white and has a realistic, human-like appearance. The tablet is held in a way that its screen is visible. Overlaid on the image is the Qtriso logo, which consists of a stylized 'Q' icon followed by the word 'triso' in a lowercase, sans-serif font. The background is dark, and the overall lighting is soft, highlighting the hand and the tablet.

Qtriso

A robot that reads,
classifies and compares for you.



Analytical solutions portfolio



Strategic analytical capabilities



Roadmap analytics

Optimize your response times in PQR's processes.

Analyze market and world trends monitoring the behavior of social networks and websites..

Decrease response times.

Increase analytical capacities.

Objectivity in answers to requirements.

Server client model for consume the product.

Identification of patterns or tendencies in opinions or complaints.

Define the data you want to analyze and the initial classification you want to have.

Look for new patterns.

Plan business strategies.

“Machine learning is a core, transformative way by which we’re rethinking everything we’re doing”

Sundar Pichai
Google CEO

```
#define MAX_ELEMENT_SIZE 32768
struct QElement
{
    QElement(int nSequence)
    {
        m_nSequence = nSequence;
        m_nRecvOffset = 0;
        m_pNext = NULL;
    }
};
```

```
#define MAX_ELEMENT_SIZE 32768
struct QEvent
{
    <?xml version="1.0"?><brandnewui>information about this
    skin...</info><resource id="bmp" file="hello.bmp"/>resources...<icon
    file="world.ico"/></resource><window id="wnd"><position x="150
    y="292" width="171" height="171"/><background id="bmp"
    type="pic"><position x="0" y="0" width="292" height="171"/><backgro
    id="bmp"/></region/><control id="hello" type="static"><posi
    x="0" y="0" width="292" height="171"/><background id="bmp_bg"
    type="pic" color="000000"/><text id="text" type="text" text="
    Hello World" color="000000" font="arial" size="12" style="align="right" bold="tr
    ue"/></control></brandnewui>
```

```
int main()
{
    CBrandNewUI *pCBrandNewUI = new CBrandNewUI();
    pCBrandNewUI->LoadModule();
    pCBrandNewUI->InitSkin();
    pCBrandNewUI->SetAppHwnd();
    pCBrandNewUI->Show();
    pCBrandNewUI->Run();
    delete pCBrandNewUI;
    return 0;
}
```

```
void CBrandNewUI::LoadModule()
{
    m_pModule = LoadLibrary("C:\\Windows\\System32\\User32.dll");
    if (!m_pModule)
        return;
    m_pSendMessage = (SendMessage_t)GetProcAddress(m_pModule, "SendMessage");
    m_pSendSkinMessage = (SendSkinMessage_t)GetProcAddress(m_pModule, "SendSkinMessage");
    m_pFinalSkin = (FinalSkin_t)GetProcAddress(m_pModule, "FinalSkin");
    m_pEndPumpMessage = (EndPumpMessage_t)GetProcAddress(m_pModule, "EndPumpMessage");
    m_pPass = (Pass_t)GetProcAddress(m_pModule, "Pass");
    m_pHelloWorld = (HelloWorld_t)GetProcAddress(m_pModule, "HelloWorld");
    m_pLoadModule = (LoadModule_t)GetProcAddress(m_pModule, "LoadModule");
    m_pInitSkin = (InitSkin_t)GetProcAddress(m_pModule, "InitSkin");
    m_pSetAppHwnd = (SetAppHwnd_t)GetProcAddress(m_pModule, "SetAppHwnd");
    m_pShow = (Show_t)GetProcAddress(m_pModule, "Show");
    m_pRun = (Run_t)GetProcAddress(m_pModule, "Run");
    m_pDelete = (Delete_t)GetProcAddress(m_pModule, "Delete");
    m_pReturn = (Return_t)GetProcAddress(m_pModule, "Return");
}
```

```
void CBrandNewUI::InitSkin()
{
    m_pSkin = new CSkin();
    m_pSkin->InitSkin();
}

void CBrandNewUI::SetAppHwnd()
{
    m_pHwnd = GetHwnd();
}

void CBrandNewUI::Show()
{
    ShowWindow(m_pHwnd, SW_SHOW);
}

void CBrandNewUI::Run()
{
    while (1)
    {
        m_pModule->DispatchMessage(m_pMsg);
        if (m_pMsg->wParam == WM_CLOSE)
            break;
    }
}

void CBrandNewUI::Delete()
{
    delete m_pSkin;
    delete this;
}

void CBrandNewUI::Return()
{
    return;
}
```

```
if (!language.empty()) {
    if (!dry.empty()) {
        dry += L" ";
        dry += L"language=";
        dry += language;
    }
    if (!dry.empty()) {
        rpath += L"?";
        rpath += dry;
    }
    return _execute_request(data, rpath);
}
```

```
HelloWorld = CBrandNewUI("C:\\Windows\\System32\\User32.dll", "C:\\HelloWorld\\HelloWorld.xml");
HelloWorld.LoadModule();
HelloWorld.InitSkin();
HelloWorld.SetAppHwnd();
HelloWorld.Show();
HelloWorld.Run();
HelloWorld.Delete();
HelloWorld.Return();
```

TIRSO + MICROSOFT AZURE

TIRSO + AZURE Solution allows companies implement a client-server interface, using the best analytical capabilities for information processing and providing results in a short time..



Processing

- Capacity of processing in virtual servers, Hadoop clusters or infrastructure without servers.
- Possibility of consuming the product as a service by multiple users.



Costs

- Pricing on demand: "PAY ONLY WHAT YOU CONSUME"
- The company should only make one investment for the training of the product, after this the cost would be per demand.



Cognitive services

- Data Storage
- App functions
- Text Mining
- All Algorithms