NTX Solusi Teknologi
Next Technology

INSIGHT AI

Knowledge modelling made simple

INSIGHT AI

Knowledge modelling made simple

Insight AI was designed to automate knowledge discovery by learning the relationships between facts that it had found. It was not designed to replace the users, but instead to help and work together with them, by continuously learning new facts from the data and reasoning the relationships between them, while continuously asking its user for feedback. This way, Insight AI does not wildly learn things out of context and proportion, but instead becomes a trusty partner that was uniquely moulded for its users.

Insight App

At the core of knowledge modelling is an application that serves as an interface between human intuition and machine knowledge discovery. Building on top of an active learning system and a knowledge management system, Insight App is the bridge that connected analysts to the entirety of knowledge contained in your organisation.

Insight App is designed to be a standalone desktop app that directly interfaced with your knowledge management system, and as such, it was built on top of several modules. Including but not limited to:

Graph Visualisation Library

The best way to model knowledge graph is by providing intuitive graph modelling techniques to users, empowering their understanding of the events that defined an issue. This is why we focus on lightweight graph visualisation libraries to allow users to quickly create graphs of entities and the events they're involved in.

Information Validation Module

Analysts have to work with extraneous information at all times. Sometimes they find information that are relevant, and sometimes less relevant. This module helps them in figuring out, and classifying, which information is more relevant, and which information requires improved validation. With this, we seek to train our system to improve its understanding of the information that users require in a given analytical context.

Case Management Module

In building an analysis, analysts need a way to organise their thoughts and ideas in a manageable way. As such, we develop a robust case management module where analysts can build their analysis from the ground up, using the information contained within our knowledge management system.

RiskMapper

There is no better way to implement a knowledge modelling module than by creating an evaluative system to figure out the relative risks of a topic. By allowing analysts to map multidimensional risks, our system created a pipeline that simplified the workflow of evaluating the risks that events and issues can contribute to.

As such, the system was built on top of three core modules:

Risk Modelling Module

Every action begets a reaction, and for our system, the reaction is the risk value of said actions. The module provides an AI-empowered risk scoring of facts given a specific context. This AI was designed to self-evaluate and actively learn about the continuously changing parameters, to increase its accuracy.

Risk Scoring Module

There is no perfect artificial intelligence. Every machine learning models require feedbacks from analysts, and so, this module was built to complement our system in allowing analysts to provide feedbacks to the machine, and allowing it to learn about the relative risks of a given fact in a given context.

Risk Validation Module

Another feedback mechanism for our active learning system is a module where analysts can directly intervene to system processes in scoring events. Whether the risk score is right or wrong, or the context is incorrect, analysts can teach the system through this module to speed up the learning process of the system.

Providence

A secondary system that supported the machine in understanding what constituted as risk and allowed the machine to predict the risk values of a fact. By incorporating this module into the system, we allowed the machine to make inferences about what it can expect as a risk and what the analysts expected to be a risk, and actively evaluate its predictions in comparison to user inputs.

This system contained three interdependent modules:

Risk Analysis Module

Given the user inputs of what constituted as risk, this system classified, clustered, and contextually learned about which facts have risk values in certain dimensions. This analysis allowed the system to finely tuned the risk factors in specific contexts tailored to analyst's requirements without complicated fine tuning required.

Anomaly Detection Module

Given possible discrepancies between risk values of a fact between one context and another, this module allowed the system to highlight the discrepancies and ask users for specific inputs in the specific contexts where the anomaly happened. This clarification would then be utilised by the system to improve its understanding about risks and their contexts.

Predictive Modelling Engine

Given the inputs about risk scores and their contexts, it's only natural that we incorporate a module that could help analysts in predicting the risk values of facts within a context. This capacity would greatly simplify user workflow in risk modelling and analysis, by taking the hard work of figuring out the values and freeing the analysts to figure out the implications of such risk values.

EventMap

With a diverse information in the knowledge base about events, facts, and their relationships, it would be troublesome should analysts have to traverse the complexity of the graph environment to understand about events contained within. That is why we seek to automate the process using three modules:

Graph Partitioning Module

This module seeks to find the most important members of a graph that relates to a fact. It created partitions, grouping the most relevant facts as an independent event that contained the actors, the location, the time frame, and other relevant facts, that analysts can use to make the set explainable as an event.

Graph Extraction Module

Given the possibility of having multiple facts relating to an event, and multiple events tied to a fact, we implemented an efficient algorithm to extract the most robust subgraph from the knowledge graph in order to represent the richest information available about a context, an issue, an event, according to user search parameters.

Event-Graph Parsing Module

Combining the outputs from the two aforementioned modules, this module wrapped an event-graph as a set containing the facts, actors, time, location, and relevant attributes that would represent any event as robustly as possible, given the information contained within the graph. This set would be presented to the analyst as a graph containing all the relevant facts, including the risk values.

Precedence

Given an event, this system seeks to make the most efficient representation of the facts and relationships contained in the event. Combining all the most relevant information from various sources related to the event, this module then constructs the most efficient graph that fits the analyst's parameters. Building upon inputs from the previous modules, this system introduced three constituent modules, explained below:

Event Detection Module

Various events can happen in one time frame, covered by various sources with different interpretations containing different facts. This module then integrated all the facts in an ontologically consistent frame to simplify analysts in providing an analysis about the facts contained within an event.

Event-Risk Modelling Engine

Attaching risk values to an event is a different process than attaching risk values to facts contained within an event. We tackled this process by creating a set of algorithms that seek to predict the risk trajectories of an event and the risks contained within the event. This process happened in conjunction with the processes happening in the aforementioned modules, enabling analysts to focus on the likelihood of event escalation or de-escalation in terms of risk dynamics.

Next-Event Prediction Engine

Given the various probable events contained in the graph, and the events that analysts had acquired from the graph, the engine seek to automatically discover the likeliest event that would follow any given event. This engine works iteratively to improve its predictive capacity and learn from the pattern of event sequences that analysts had made.

Infomap

Providing additional information on demand, the module was designed to ingest and integrate information contained in structured data formats to enrich the facts that analysts queried. This module is a customisable, lightweight microservice that allowed for data enrichment in-time to analyst's requests.

Demographic Data Ingestion

Made for consistently structured datasets in static tables, this module works best for demographic data, providing integrated information about people and their static information. Given a static table, the module indexed and ingested the data contained in the tables for quick display on demand.

Graph Extraction Module

Whereas the previous module works best for static datasets, given the nature of telecommunication data, we developed this module to ingest streaming data that continuously update itself with the most recent information. This microservice module seek to bridge the need for streaming datasets without affecting the whole AI system, reducing the information overload both for the system and for your analysts.

NTX Solusi Teknologi

Mega Plaza 4th Floor
Jl. HR. Rasuna Said Kav C-3 Jakarta 12920, Indonesia
Tel. (62-21) 5204873 Fax. (62-21) 5204874