Automating the junior analyst

Cyber security report generation with classic AI

by Sergey Polzunov

BlackStork

Effective reporting is difficult

Requirements

- **Efficiency**: there is a demand for extensive communication towards both internal and external stakeholders, placing a significant burden on the security teams to produce a wide range of reports.
- **Quality**: inconsistent quality hinders understand within an organization. With reports varying in accuracy and clarity, it is difficult for stakeholders to make informed decisions promptly.
- **Customisation**: crafting reports to meet the specific needs of various stakeholders manually requires time and effort.

The problem with efficiency

- **Efficiency**: it is difficult to scale up the report production:
  - while creative in-depth research work enjoys a dedicated time budget, simpler reports are often neglected and left to junior analysts to do
  - the reporting features in the security platforms are simplistic: mostly multi-page text-area forms or WYSIWYG editors *(bonus points if you can insert pictures or tables!)*

The problem with quality

- **Quality**: the teams develop internal guidelines and templates but there are associated costs:
  - template management is painful – the templates are treated as artifacts and are stored in Google Docs, OneDrive, Confluence, or in endless email threads with Word documents *(bonus points if you use git!)*
  - there are no built-in control checks to make sure all required data is included in the final text
  - stylistic and formatting constraints are not applied automatically, leaving space for inconsistencies.

BlackStork

The problem with customisation

- **Customisation**: the final form of the report depends on the target audience *and* on the input data available:
  - the multiple variants of the same report are either produced from a clean slate or require some creative frankenstein-ing of various other templates into one
  - writer’s block is still an issue when starting from scratch
  - the templates are rigid and must be adjusted manually if the shape of input data changes

Let’s automate it!

- **Efficiency**: using code to generate the reports
- **Quality** and **Customisation**: introducing “reports-as-code” templating:
  - A template is specification tree written in a custom declarative DSL
  - A template defines the document structure and the data needed to populate the document, without prescribing the exact words / phrases to be used
  - The same template can be used to produce multiple variants of the report or produce the report from different shapes of input data:
    - the “compilation” step of combining the **template**, the **configuration**, and the **input data** defines the final form of the document.

BlackStork

Let’s automate it!

! Full automation is unachievable – somebody always needs to perform the last editing step and sign off on the final product

- **Efficiency**: using code to generate the reports
- **Quality** and **Customisation**: introducing “reports-as-code” templating:
    - A template is specification tree written in a custom declarative DSL
    - A template defines the document structure and the data needed to populate the document, without prescribing the exact words / phrases to be used
    - The same template can be used to produce multiple variants of the report or produce the report from different shapes of input data:
        - the “compilation” step of combining the **template**, the **configuration**, and the **input data** defines the final form of the document.

Where is AI here?

Classic and modern AI from a bird's-eye view

- Symbolic AI
  - Rule-based systems that relied on predefined rules and logic, making them suitable for deterministic well-defined tasks and problems.
  - Excels in the domains where the knowledge can be structured and codified into a clean input data. No learning ability.
- ML models
  - Utilise machine learning algorithms to enable systems to learn complex patterns, relationships, and behaviours from data. Training requires vast amounts of training data.

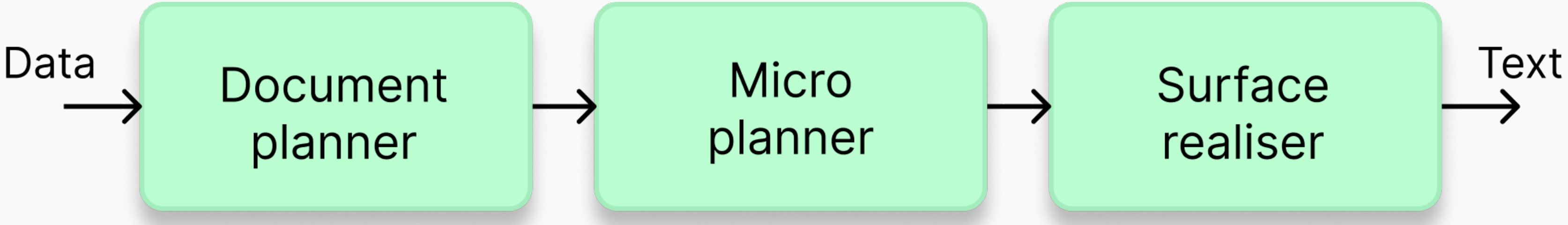
Why not just throw a LLM at it?

Using LLMs as end-to-end solution relies on LLM as both *a knowledge base* and *a comprehension engine*:

- there are issues with the embedded knowledge base:
    - limited domain expertise
    - over-reliance on training data, with all its built-in biases
- there are issues with the comprehension engine:
    - limited context understanding ability
    - absence of transparency, explainability and auditability (cyber security becoming more regulated!) that hinders QA efforts and erodes trust

Natural Language Generation pipeline

Natural Language Generation pipeline



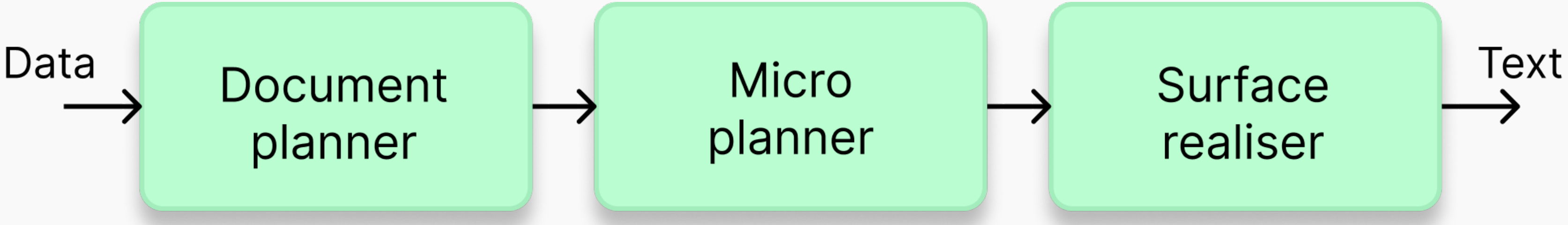
- selects the template and the relevant input data
- defines the document structure based on configuration and available data

- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

-generates the final text from a spec
-performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

BlackStork

Natural Language Generation pipeline



Document planner

- selects the template and the relevant input data
- defines the document structure based on configuration and available data

Micro planner

- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

Surface realiser

- generates the final text from a spec
- performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

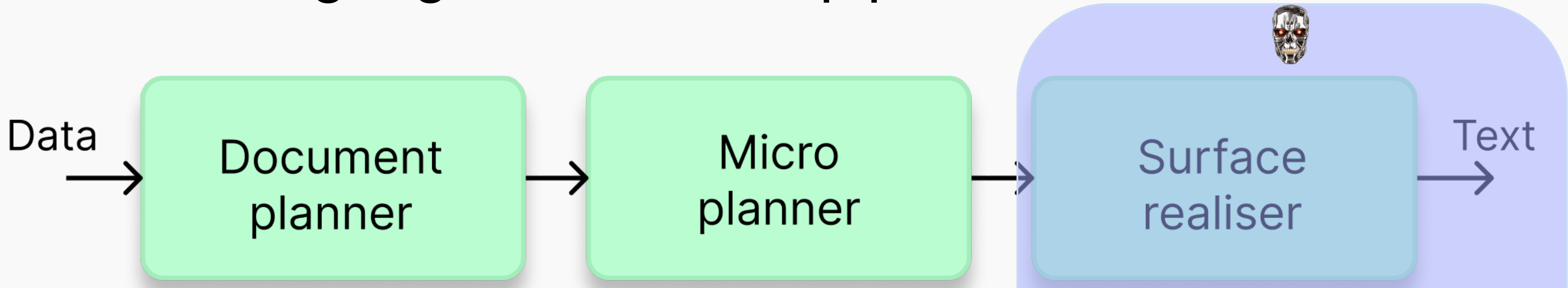
Threat actor X
- associated TTPs
- **associated campaigns**
- associated victims

Threat actor X
- **created by: Vendor V**
- **aliases: Alias1, Alias2**

Associated **Campaign A**
- **first seen: Oct 2, 2023**

Threat actor X was described by **Vendor V.** The actor is also known as "**Alias1**" and "**Alias2**". The actor is associated with **Campaign A,** first observed on **Oct 2, 2023.**

Natural Language Generation pipeline with ML



Document planner

- selects the template and the relevant input data
- defines the document structure based on configuration and available data

Micro planner

- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

Surface realiser

-generates the final text from a spec
-performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

Threat actor X
- associated TTPs
- **associated campaigns**
- associated victims

→

Threat actor X
- **created by: Vendor V**
- **aliases: Alias1, Alias2**

Associated **Campaign A**
- **first seen: Oct 2, 2023**

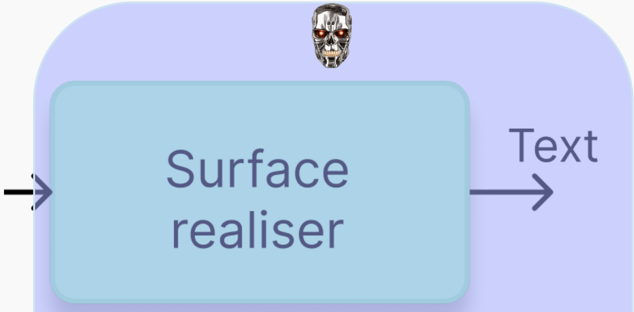
→

Threat actor X was described by **Vendor V.** The actor is also known as “**Alias1**” and “**Alias2**”.
The actor is associated with **Campaign A,** first observed on **Oct 2, 2023.**

Natural Language Generation pipeline with ML

Narrow scope allows for specialised ML models:

- Custom RNN data-to-text / graph-to-text models
- BART language model (2018): converting a partial set of unordered non-inflected tokens into a full set of ordered inflected tokens



-generates the final text from a spec
-performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

Threat actor X was described by **Vendor V.** The actor is also known as "**Alias1**" and "**Alias2**".
The actor is associated with **Campaign A**, first observed on **Oct 2, 2023.**

- *Yao Zhou, Cong Liu, and Yan Pan. 2016. "Modelling sentence pairs with tree-structured attentive encoder"*
- *Farhood Farahnak, Laya Rafiee, Leila Kosseim and Thomas Fevens. 2020. "Surface Realization Using Pretrained Language Models"*

Natural Language Generation pipeline with ML

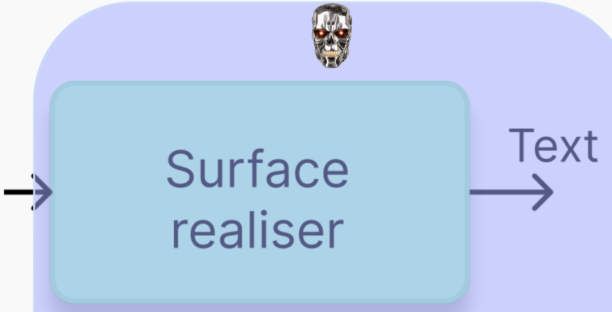
Narrow scope allows for specialised ML models:

- Custom RNN data-to-text / graph-to-text models
- BART language model (2018): converting a partial set of unordered non-inflected tokens into a full set of ordered inflected tokens

or a **constrained** use of LLMs (GPT, LLaMA, etc).

The model just needs to know English language.

The final editing step is the smallest.



-generates the final text from a spec
-performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

Threat actor X was described by **Vendor V.** The actor is also known as "**Alias1**" and "**Alias2**".
The actor is associated with **Campaign A,** first observed on **Oct 2, 2023.**

- *Yao Zhou, Cong Liu, and Yan Pan. 2016. "Modelling sentence pairs with tree-structured attentive encoder"*
- *Farhood Farahnak, Laya Rafiee, Leila Kosseim and Thomas Fevens. 2020. "Surface Realization Using Pretrained Language Models"*

Natural Language Generation pipeline with ML

Data → Document planner → Micro planner → Surface realiser → Text

Document planner
- selects the template and the relevant input data
- defines the document structure based on configuration and available data

Micro planner
- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

Surface realiser
-generates the final text from a spec
-performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

Threat actor X
- associated TTPs
- **associated campaigns**
- associated victims

→

"Threat actor X" entity
- **created by: Vendor V**
- **aliases: Alias1, Alias2**

"**Campaign A**" entity
- **first seen: Oct 2, 2023**

→

Threat actor X was described by **Vendor V.** The actor is also known as "**Alias1**" and "**Alias2**". The actor is associated with **Campaign A,** first observed on **Oct 2, 2023.**

Natural Language Generation pipeline with ML

Data → Document planner → Micro planner → Surface realiser → Text

Document planner
- selects the template and the relevant input data
- defines the document structure based on configuration and available data

Micro planner
- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

Surface realiser
- generates the final text from a spec
- performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

```
Threat actor X
 - associated TTPs
 - associated campaigns
 - associated victims
```

→

```
"Threat actor X" entity
 - created by: Vendor V
 - aliases: Alias1, Alias2
"Campaign A" entity
 - first seen: Oct 2, 2023
```

→

Threat actor X was described by **Vendor V.** The actor is also known as "**Alias1**" and "**Alias2**". The actor is associated with **Campaign A,** first observed on **Oct 2, 2023.**

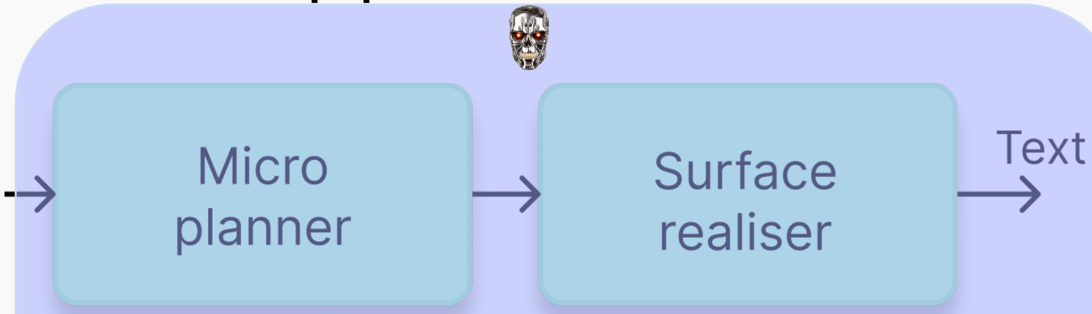
Natural Language Generation pipeline with ML

Wider scope requires the model to "understand":

- summarisation
- rephrasing

Less control:

- it is difficult to impose restrictions on the output
- the result text is less predictable
- larger editing step



- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

-generates the final text from a spec
-performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

"Threat actor X" entity
- **created by: Vendor V**
- **aliases: Alias1, Alias2**

"**Campaign A**" entity
- **first seen: Oct 2, 2023**

→

Threat actor X was described by **Vendor V.** The actor is also known as "**Alias1**" and "**Alias2**".
The actor is associated with **Campaign A,** first observed on **Oct 2, 2023.**

Natural Language Generation pipeline with ML

Data → **Document planner** → **Micro planner** → **Surface realiser** → Text

Document planner
- selects the template and the relevant input data
- defines the document structure based on configuration and available data

Micro planner
- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

Surface realiser
- generates the final text from a spec
- performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

```
Threat actor X
 - associated TTPs
 - associated campaigns
 - associated victims
```

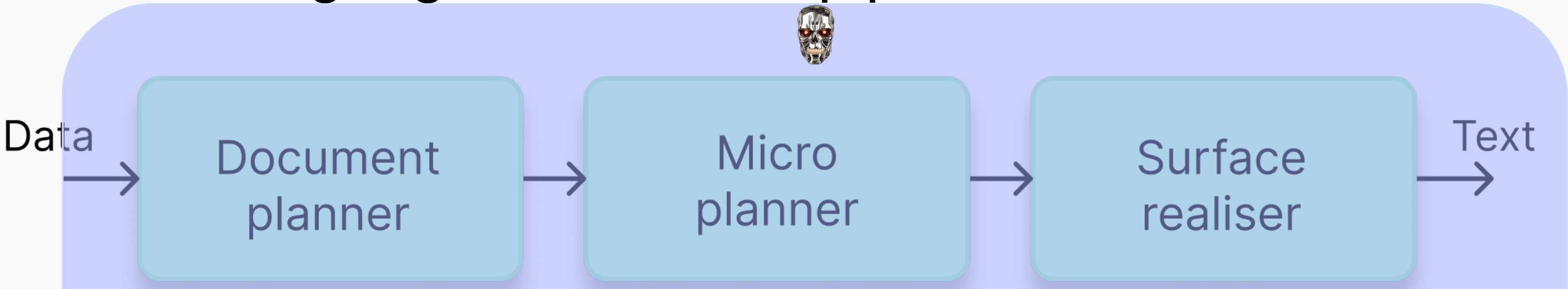
→

```
"Threat actor X" entity
 - created by: Vendor V
 - aliases: Alias1, Alias2
"Campaign A" entity
 - first seen: Oct 2, 2023
```

→

```
Threat actor X was described
by Vendor V. The actor is also
known as "Alias1" and
"Alias2".
The actor is associated with
Campaign A, first observed on
Oct 2, 2023.
```

Natural Language Generation pipeline with ML



- selects the template and the relevant input data
- defines the document structure based on configuration and available data

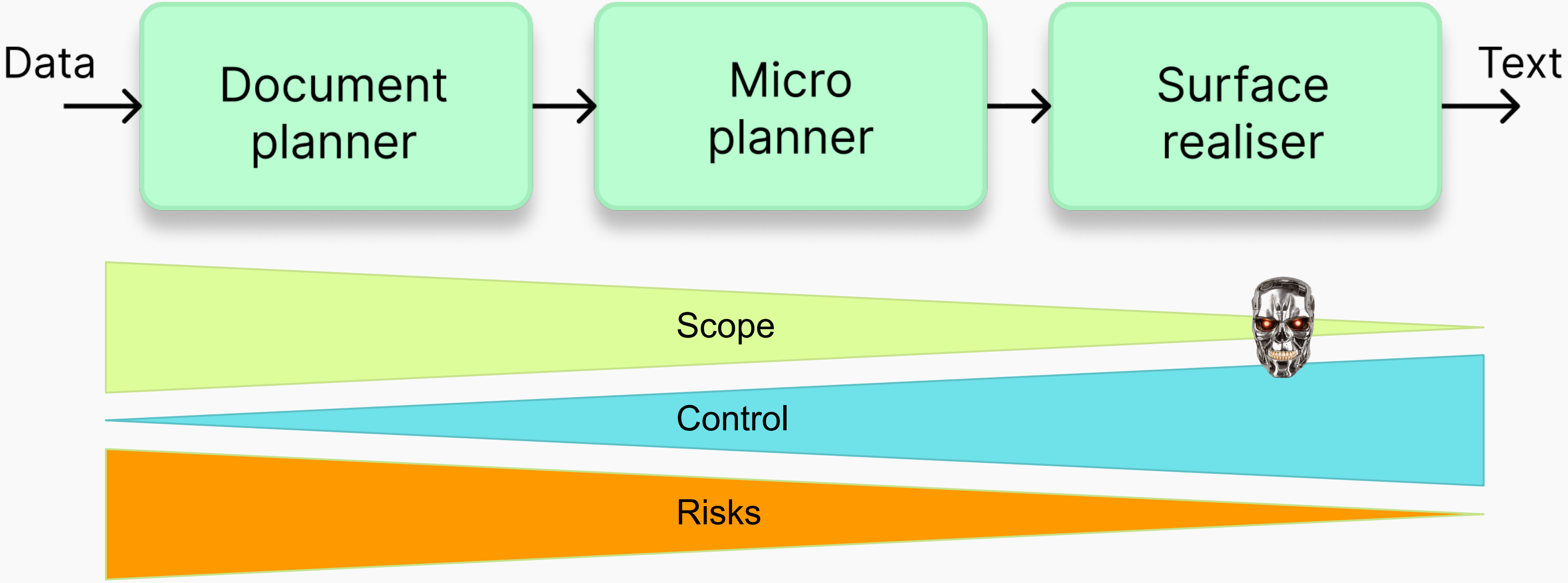
- selects properties and data points from the input data
- defines the aggregations of the data points to be expressed together

-generates the final text from a spec
-performs syntactical, morphological and orthographic realisation (SimpleNLG, Grammatical Framework, etc)

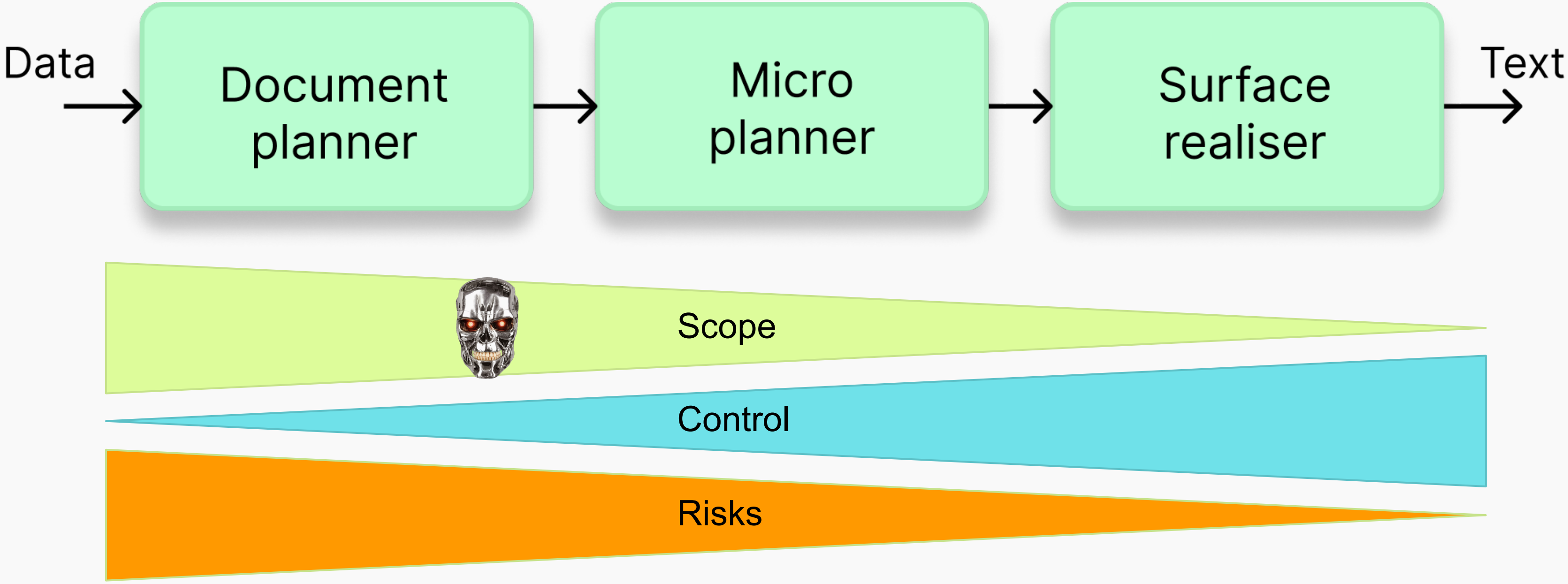
Let's throw LLM at it!

Lowest level of control, the highest risk, largest editing step.

The right amount of ML in NLG



The right amount of ML in NLG



Conclusions

- **Re-evaluate existing tech:**
    - classic rule-based systems have a lot to offer and can be nicely combined with the creativity of the modern ML models
- **Define how much ML is right for you:**
    - be mindful of the risks that come with deploying ML models (and LLMs) in production in mission critical workflows
- **Sometimes less is more:**
    - LLMs are powerful and easy to integrate with, but there are smaller ML models that might fit your use cases better, are easier to manage and can be run on-prem
- **Be on top of your reporting game!**
    - NIS2 brings additional pressure to have the communications streamlined

Thank you!

sergey@blackstork.io

https://blackstork.io