Natural Language Processing and Machine Learning in AllegroGraph

The majority of our customers build Knowledge Graphs with Natural Language and Machine learning components. Because of this trend AllegroGraph now offers strong support for the use of Natural Language Processing and Machine learning.

Franz Inc has a team of NLP engineers and Taxonomy experts that can help with building turn-key solutions. In general however, our customers already have some expertise in house. In those cases we train customers in how to take the output of NLP and ML processing and turn that into an efficient Knowledge Graph based on best practices in the industry.

This document primarily describes the NLP and ML plug-in AllegroGraph.

Note that many enterprises already have a data science team with NLP experts that use modern open source NLP tools like Spacy, Gensim or Polyglot, or Machine Learning based NLP tools like BERT and Scikit-Learn. In another blog about Document Handling we describe a pipeline of how to deal with NLP in Document Knowledge Graphs by using our NLP and ML plugin and mix that with open source tools.

PlugIn features for Natural Language Processing and Machine Learning in AllegroGraph.

Here is the outline of the plugin features that we are going to describe in more detail.

Machine learning

- data acquisition
- classifier training

- feature extraction support
- performance analysis
- model persistence

NLP

- handling languages
- handling dictionaries
- tokenization
- entity extraction
- Sentiment analysis
- basic pattern matching

SPARQL Access

- Future development

Machine Learning

ML: Data Acquisition

Given that the NLP and ML functions operate within AllegroGraph, after loading the plugins, data acquisition can be performed directly from the triple-store, which drastically simplifies the data scientist workflow. However, if the data is not in AllegroGraph yet we can also import it directly from ten formats of triples or we can use our additional capabilities to import from CSV/JSON/JSON-LD.

Part of the Data Acquisition is also that we need to pre-process the data for training so we provide these three functions:

- prepare-training-data
- split-dev-test
- equalize (for resampling)

Machine Learning: Classifiers

- Currently we provide simple linear classifiers. In case

there's a need for neural net or other advanced classifiers, those can be integrated on-demand.
- We also provide support for online learning (online machine learning is an ML method in which data becomes available in a sequential order and is used to update the best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once). This feature is useful for many real-world data sets that are constantly updated.
- The default classifiers available are Averaged Perceptron and AROW

Machine Learning: Feature Extraction

Each classifier is expecting a vector of features: either feature indices (indicative features) or pairs of numbers (index – value). These are obtained in a two-step process:

1. A classifier-specific extract-features method should be defined that will return raw feature vector with features identified by strings of the following form: prefix|feature.

The prefix should be provided as a keyword argument to the collect-features method call, and it is used to distinguish similar features from different sources (for instance, for distinct predicates).

2. Those features will be automatically transformed to unique integer ids. The resulting feature vector of indicator features may look like the following: #(1 123 2999 …)

Note that these features may be persisted to AllegroGraph for repeated re-use (e.g. for experimenting with classifier hyperparameter tuning or different classification models).

Many possible features may be extracted from data, but there is a set of common ones, such as:

1. individual tokens of the text field
2. ngrams (of a specified order) of the text field
3. presence of a token in a specific dictionary (like, the dictionary of slang words)
4. presence/value of a certain predicate for the subject of the current triple
5. length of the text

And in case the user has a need for special types of tokens we can write specific token methods, here is an example (in Lisp) that produces an indicator feature of a presence of emojis in the text:

```
(defmethod collect-features ((method (eql :emoji)) toks &key
pred)
(dolist (tok toks)
(when (some #'(lambda (code)
  (or (<= #x1F600 code #x1F64F)
      (<= #x1F650 code #x1F67F)
      (<= #x1F680 code #x1F6FF)))
   (map 'vector #'char-code tok))
(return (list "emoji")))))
```

Machine Learning: Integration with Spacy

The NLP and ML community invents new features and capabilities at an incredible speed. Way faster than any database company can keep up with. So why not embrace that? Whenever we need something that we don’t have in AllegroGraph yet we can call out to Spacy or any other external NLP tool. Here is an example of using feature extraction from Spacy to collect indicator features of the text dependency parse relations:

```
(defmethod collect-features ((method (eql :dep)) deps &key
pred dep-type dep-labels)
 (loop :for ds :in deps :nconc
  (loop :for dep :in ds
```

```
   :when (and (member (dep-tag dep) dep-labels)
              (dep-head dep)
              (dep-tok dep))
    :collect (format nil "dep|~a|~a_~a"
              dep-type
              (tok-word (dep-head dep)
              (tok-word (dep-tok dep))))))
```

The demonstrated integration uses Spacy Docker instance and its HTTP API.

Machine Learning: Classifier Analysis

We provide all the basic tools and metrics for classifier quality analysis:

- accuracy
- f1, precision, recall
- confusion matrix
- and an aggregated classification report

Machine Learning: Model Persistence

The idea behind model persistence is that all the data can be stored in AllegroGraph, including features and classifier models. AllegroGraph stores classifiers directly as triples. This is a far more robust and language-independent approach than currently popular among data scientists reliance on Python pickle files. For the storage we provide a basic triple-based format, so it is also possible to interchange the models using standard RDF data formats.

The biggest advantage of this approach is that when adding text to AllegroGraph we don’t have to move the data externally to perform the classification but can keep the whole pipeline entirely internal.

Natural Language Procession (NLP)

NLP: Language Packs

Most of the NLP tools are language-dependent: i.e. there’s a general function that uses language-specific model/rules/etc. In AllegroGraph, support for particular languages is provided on-demand and all the language-specific is grouped in the so called “language pack” or langpack, for short — a directory with a number of text and binary files with predefined names.

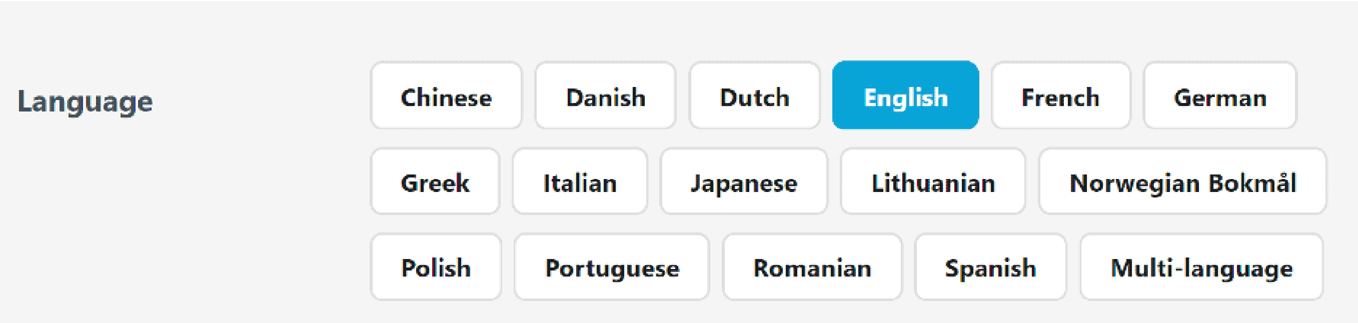
Currently, the langpack for English is provided at nlp/langs/en.zip, with the following files:

- contractions.txt — a dictionary of contractions
- abbrs.txt — a dictionary of abbreviations
- stopwords.txt — a dictionary of stopwords
- pos-dict.txt — positive sentiment words
- neg-dict.txt — negative sentiment words
- word-tok.txt — a list of word tokenization rules

Additionally, we use a general dictionary, a word-form dictionary (obtained from Wiktionary), and custom lexicons.

Loading a langpack for a particular language is performed using load-langpack.

Creating a langpack is just a matter of adding the properly named files to the directory and can be done manually. The names of the files should correspond to the names of the dictionary variables that will be filled by the pack. The dictionaries that don’t have a corresponding file will be just skipped.We have just finished creating a langpack for Spanish and it will be published soon. In case you need other dictionaries we use our AG/Spacy infrastructure. Spacy recently added a comprehensive list of new languages:



NLP: Dictionaries

Dictionaries are read from the language packs or other sources and are kept in memory as language-specific hash-tables. Alongside support for storing the dictionaries as text files, there are also utilities for working with them as triples and putting them into the triple store.

Note that we at Franz Inc specialize in Taxonomy Building using various commercial taxonomy building tools. All these tools can now export these taxonomies as a mix of SKOS taxonomies and OWL. We have several functions to read directly from these SKOS taxonomies and turn them into dictionaries that support efficient phrase-level lookup.

NLP: Tokenization

Tokenization is performed using a time-proven rule-based approach. There are 3 levels of tokenization that have both a corresponding specific utility function and an :output format of the tokenize function:

> :parags – splits the text into a list of lists of tokens for paragraphs and sentences in each paragraph
> :sents – splits the text into a list of tokens for each sentence
> :words – splits the text into a plain list of tokens

Paragraph-level tokenization considers newlines as paragraph delimiters. Sentence-level tokenization is geared towards western-style writing that uses dot and other punctuation marks to delimit sentences. It is, currently, hard-coded, but if the need arises, additional handling may be added for other writing systems. Word-level tokenization is performed using a language-specific set of rules.

NLP: Entity Extraction

Entity extraction is performed by efficient matching (exactly or fuzzy) of the token sequences to the existing dictionary structure.

It is expected that the entities come from the triple store and there's a special utility function that builds lookup dictionaries from all the triples of the repository identified by certain graphs that have a skos:prefLabel or skos:altLabel property. The lookup may be case-insensitive with the exception of abbreviations (default) or case-sensitive.

Similar to entity extraction, there's also support for spotting sentiment words. It is performed using the positive/negative words dictionaries from the langpack.

One feature that we needed to develop for our customers is 'heuristic entity extraction' . In case you want to extract complicated product names from text or call-center conversations between customers and agents you run into the problem that it becomes very expensive to develop altLabels in a taxonomy tool. We created special software to facilitate the automatic creation of altlabels.

NLP: Basic Pattern Matching for relationship and event detection

Getting entities out of text is now well understood and supported by the software community. However, to find complex concepts or relationships between entities or even events is way harder and requires a flexible rule-based pattern matcher. Given our long time background in Lisp and Prolog one can imagine we created a very powerful pattern matcher.

SPARQL Access

Currently all the features above can be controlled as stored procedures or using Lisp as the command language. We have a new (beta) version that uses SPARQL for most of the control. Here are some examples. Note that fai is a magic-property namespace for “AI”-related stuff and inc is a custom namespace of an imaginary client:

1. Entity extraction

```
select ?ent {
   ?subj fai:entityTaxonomy inc:products .
   ?subj fai:entityTaxonomy inc:salesTerms .
   ?subj fai:textPredicate inc:text .
   ?subj fai:entity(fai:language "en", fai:taxonomy inc:products) ?ent .
}
```

The expressions ?subj fai:entityTaxonomy inc:poducts and ?subj fai:entityTaxonomy inc:salesTerms specify which taxonomies to use (the appropriate matchers are cached).
The expression ?subj fai:entity ?ent will either return the already extracted entities with the specified predicate (fai:entity) or extract the new entities according to the taxonomies in the texts accessible by fai:textPredicate.

2. fai:sentiment will return a single triple with sentiment score:

```
select ?sentiment {
   ?subj fai:textPredicate inc:text .
   ?subj fai:sentiment ?sentiment .
   ?subj fai:language "en" .
   ?subj fai:sentimentTaxonomy franz:sentiwords .
}
```

3. Text classification:

Provided inc:customClassifier was already trained previously, this query will return labels for all texts as a result of classification.

```
select ?label {
?subj fai:textPredicate inc:text .
?subj fai:classifier inc:customClassifier .
?subj fai:classify ?label .
?label fai:storeResultPredicate inc:label .
}
```

Further Development

Our team is currently working on these new features:

- A more accessible UI (python client & web) to facilitate NLP and ML pipelines
- Addition of various classifier models
- Sequence classification support (already implemented for a customer project)
- Pre-trained models shipped with AllegroGraph (e.g. English NER)
- Graph ML algorithms (deepwalk, Google Expander)
- Clustering algorithms (k-means, OPTICS)

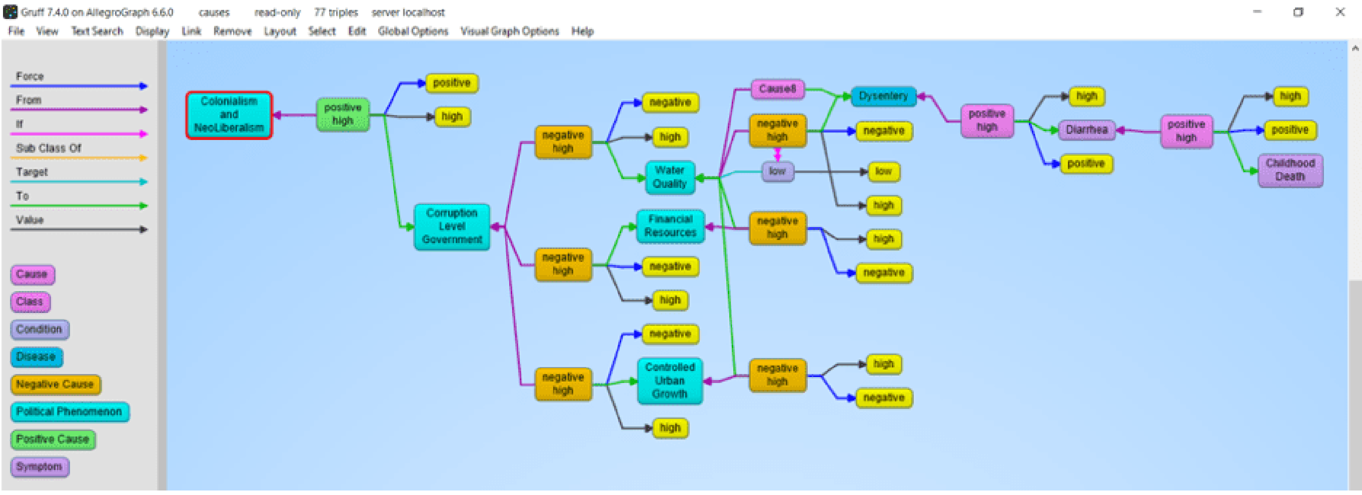
Document Knowledge Graphs with NLP and ML

A core competency for Franz Inc is turning text and documents into Knowledge Graphs (KG) using Natural Language Processing (NLP) and Machine Learning (ML) techniques in combination with AllegroGraph. In this document we discuss how the techniques described in [NLP and ML components of AllegroGraph] can be combined with popular software tools to create a robust Document Knowledge Graph pipeline.

We have applied these techniques for several Knowledge Graphs but in this document we will primarily focus on three completely different examples that we summarize below. First is the Chomsky Legacy Project where we have a large set of very dense documents and very different knowledge sources, Second is a knowledge graph for an intelligent call center where we have to deal with high volume dynamic data and real-time decision support and finally, a large government organization where it is very important that people can do a semantic search against documents and policies that steadily change over time and where it is important that you can see the history of documents and policies.

Example [1] Chomsky Knowledge Graph
The Chomsky Legacy Project is a project run by a group of admirers of Noam Chomsky with the primary goal to preserve all his written work, including all his books, papers and interviews but also everything written about him. Ultimately students, researchers, journalists, lobbyists, people from the AI community, and linguists can all use this knowledge graph for their particular goals and questions.

The biggest challenges for this project are finding causal relationships in his work using event and relationship extraction. A simple example we extracted from an author quoting Chomsky is that neoliberalism ultimately causes childhood death.



Example 2: N3 Results and the Intelligent Call Center

This is a completely different use case (See a recent KMWorld Articlehttps://allegrograph.com/knowledge-graphs-enhance-customer-experience-through-speed-and-accuracy/). Whereas the previous use case was very static, this one is highly dynamic. We analyze in real-time the text chats and spoken conversations between call center agents and customers. Our knowledge graph software provides real-time decision support to make the call center agents more efficient. N3 Results helps big tech companies to sell their high tech solutions, mostly cloud-based products and services but also helps their clients sell many other technologies and services.

The main challenge we tackle is to really deeply understand what the customer and agent are talking about. None of this can be solved by only simple entity extraction but requires elaborate rule-based and machine learning techniques. Just to give a few examples. We want to know if the agent talked about their most important talking points: that is, did the agent ask if the customer has a budget, or the authority to make a decision or a timeline about when they need the new technology or whether they actually have expressed their need. But also whether the agent reached the right person, and whether the agent talked about the follow-up. In addition, if the customer talks about competing technology we need to recognize that and provide the agent in real-time with a battle card specific to the competing technology. And in order to be able to do the latter, we also analyzed the complicated marketing materials of the clients of N3.

Example 3: Complex Government Documents

Imagine a regulatory body with tens of thousands of documents. Where nearly every paragraph has reference to other paragraphs in the same document or other documents and the documents change over time. The goal here is to provide the end-users in the government with the right document given their current task at hand. The second goal is to keep track of all the changes in the documents (and the relationship between documents) over time.

The Document to Knowledge Graph Pipeline

Process Name	Input	Output
1. Custom Taxonomy Creation	Corpus Analytics, Taxonomy tool	A SKOS taxonomy containing concepts, concept hierarchy, prefLabels, altLabels.
2. Document Preparation	Documents (pdf, word, ppt, xlsx), Apache Tika, Spacy for XML cleanup	An XML version of each document
3. Extract Document Meta Data	Document + Apache Tika	JSON dictionary of the Document MetaData
4. XML-to-Triples	XML+JSON dictionary, XMLToTriples.py	Graph-based document tree with chapters, sections, and paragraphs as triples. Also includes meta data as triples
5. Entity-Extraction	Paragraphs + taxonomies + AllegroGraph Entity extract or external extractors	Concepts, persons, places, currencies. Connected to paragraphs
6. LOD Enrichment	Paragraphs + IBM Natural Language Understanding.	Concept categories and links to DBpedia and GeoNames, etc.
7. Complex Relationship and Event extraction.	Paragraphs + Taxonomy + Rules in Spacy or AllegroGraph	Complex events and relationships, References to other document sections.
8. NLP and ML	Chapters and paragraphs + all the tools described [here], but also using Spacy, Gensim, BERT, SciKit Learn.	Similarities, sentiment, query answering, smart search, text classification, word embeddings, abstracts
9. Versioning and Document tracking	Old + New document, compare.py	Old document in historic repository, new document in current, changed graph.
10. Statistical Relationships	Concepts + OddRatio.py or OddsRatio.cl	Statistical relationships between concepts.

Let us first give a quick summary in words of how we turn documents into a Knowledge Graph.

[1] Taxonomy Creation

Taxonomy of all the concepts important to the business using open source or commercial taxonomy builders. An available industry taxonomy is a good starting point for additional customizations.

[2] Document Preparation

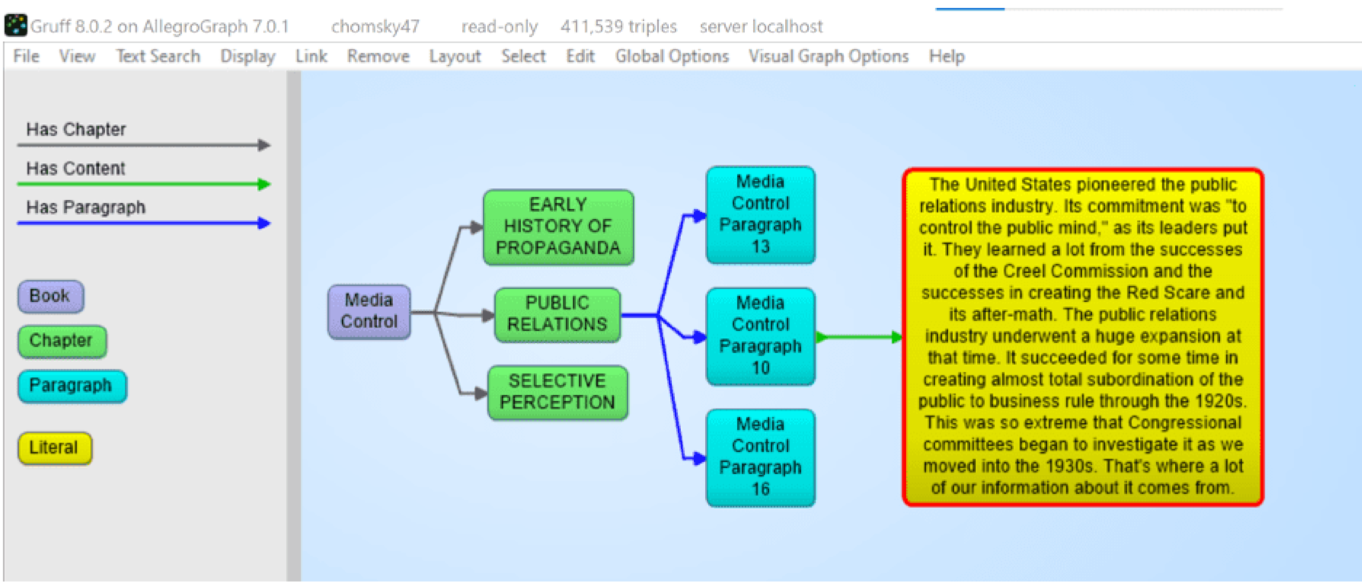
We then take a document and turn it into an intermediate XML using Apache Tika. Apache Tika supports more than 1000 document types and although Apache Tika is a fantastic tool, the output is still usually not clean enough to create a graph from, so we use Spacy rules to clean up the XML to make it as uniform as possible.

[3] Extract Document MetaData

Most documents also contain document metadata (author, date, version, title, etc) and Apache Tika will also deliver the metadata for a document as a JSON object.

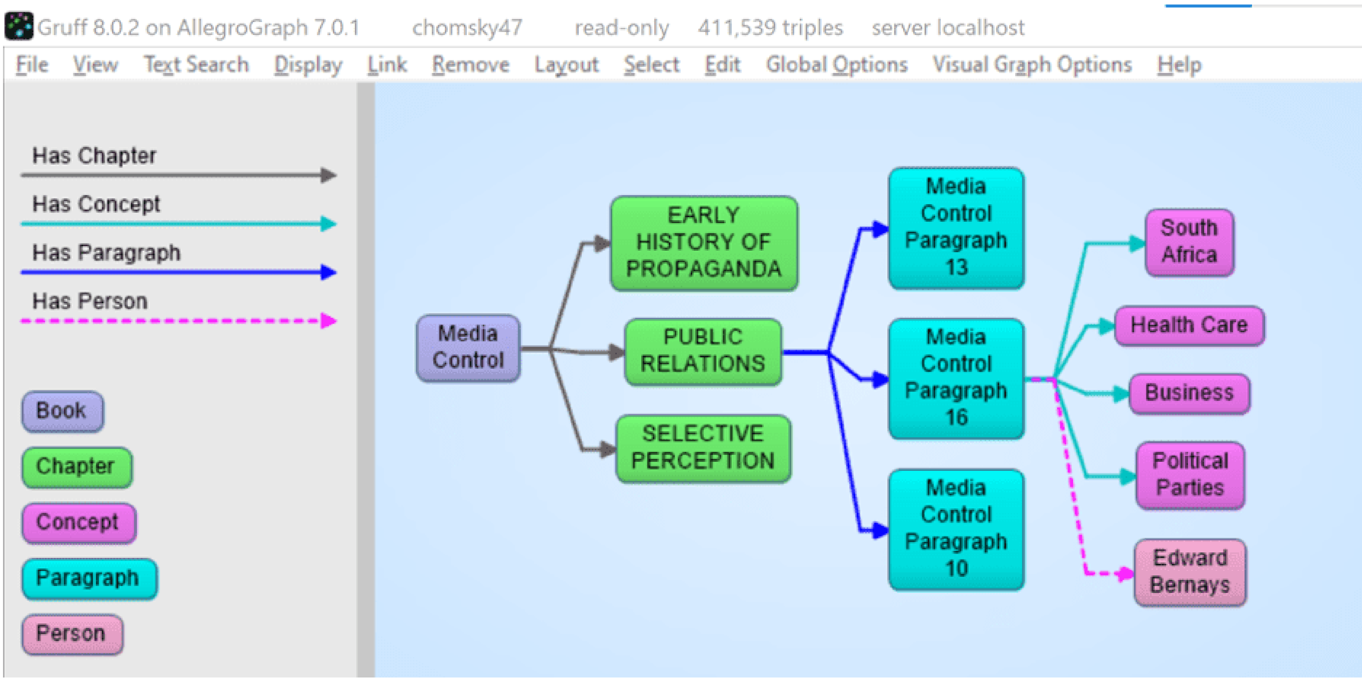
[4] XML to Triples

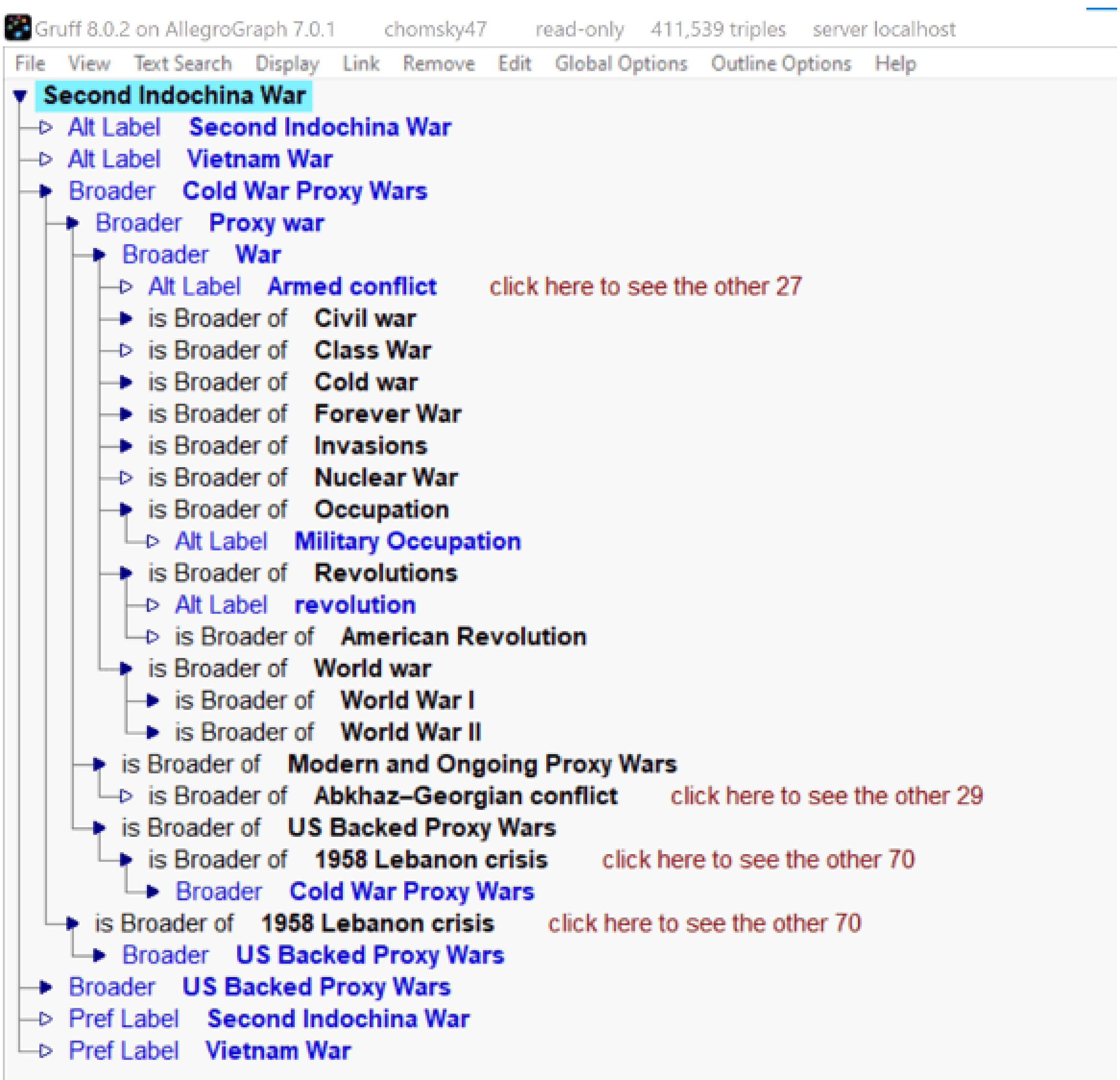
Our tools ingest the XML and metadata and transform that into a graph-based document tree. The document is the root and from that, it branches out into chapters, optionally sections, all the way down to paragraphs. The ultimate text content is in the paragraphs. In the following example we took the XML version of Noam Chomsky's book Media Control and turned that into a tree. The following shows a tiny part of that tree. We start with the Media Control node, then we show three (of the 11) chapters, for one chapter we show three (of the 6) paragraphs, and then we show the actual text in that paragraph. We sometimes can go even deeper to the level of sentences and tokens but for most projects that is overkill.



[5] Entity Extractor

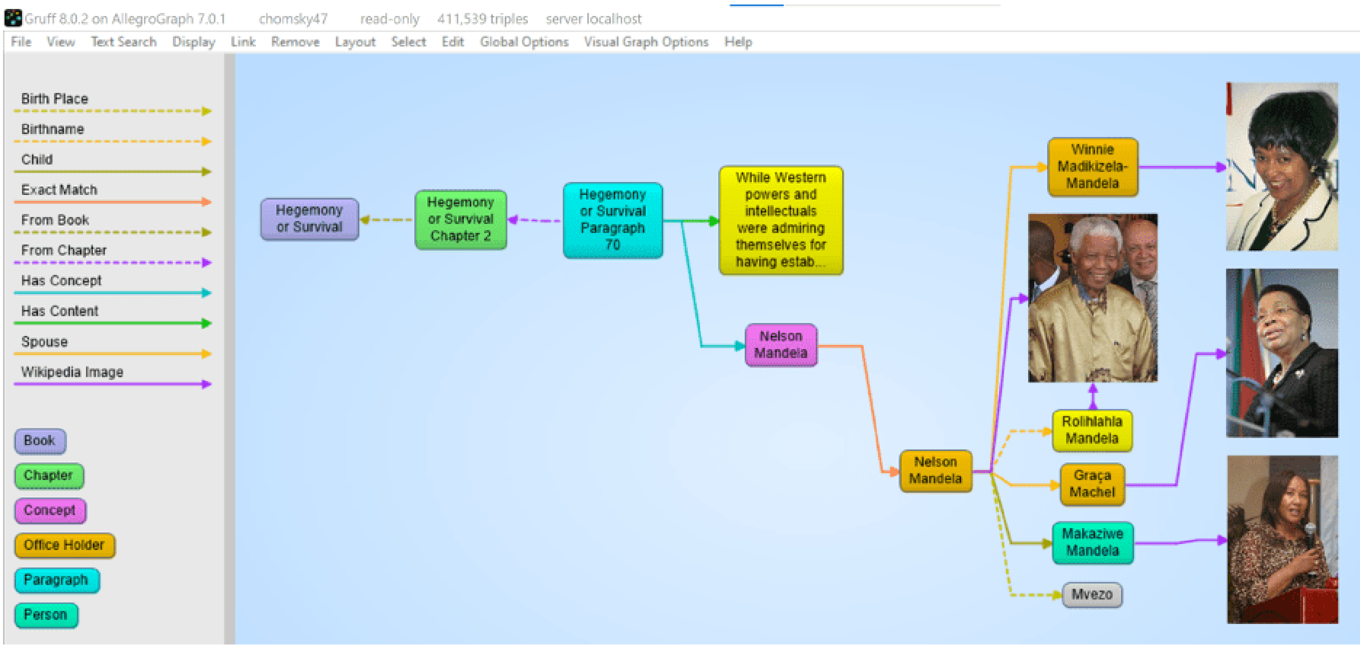
AllegroGraph's entity extractor takes as input the text of each paragraph in the document tree and one or more of the taxonomies and returns recognized SKOS concepts based on prefLabels and altLabels. AllegroGraph's entity extractor is state of the art and especially powerful when it comes to complex terms like product names. We find that in our call center a technical product name can sometimes have up to six synonyms or very specific jargon. For example the Cisco product Catalyst 9000 will also be abbreviated as the cat 9k. Instead of developing altLabels for every possible permutation that human beings *will* use, we have specialized heuristics to optimize the yield from the entity extractor. The following picture shows 4 (of the 14) concepts discovered in paragraph 16. Plus one person that was extracted by IBM's NLU.





[6] Linked Data Enrichment

In many use cases, AllegroGraph can link extracted entities to concepts in the linked data cloud. The most prominent being DBpedia, wikidata, the census database, GeoNames, but also many Linked Open Data repositories. One tool that is very useful for this is IBM's Natural Language Understanding program but there are others available. In the following image we see that the Nelson Mandela entity (Red) is linked to the dbpedia entity for Nelson Mandela and that then links to the DBpedia itself. We extracted some of his spouses and a child with their pictures.



[7] Complex Relationship and Event Extraction

Entity extraction is a first good step to 'see' what is in your documents but it is just the first step. For example: how do you find in a text whether company C1 merged with company C2. There are many different ways to express the fact that a company fired a CEO. For example: Uber got rid of Kalanick, Uber and Kalanick parted ways, the board of Uber kicked out the CEO, etc. We need to write explicit symbolic rules for this or we need a lot of training data to feed a machine learning algorithm.

[8] NLP and Machine Learning

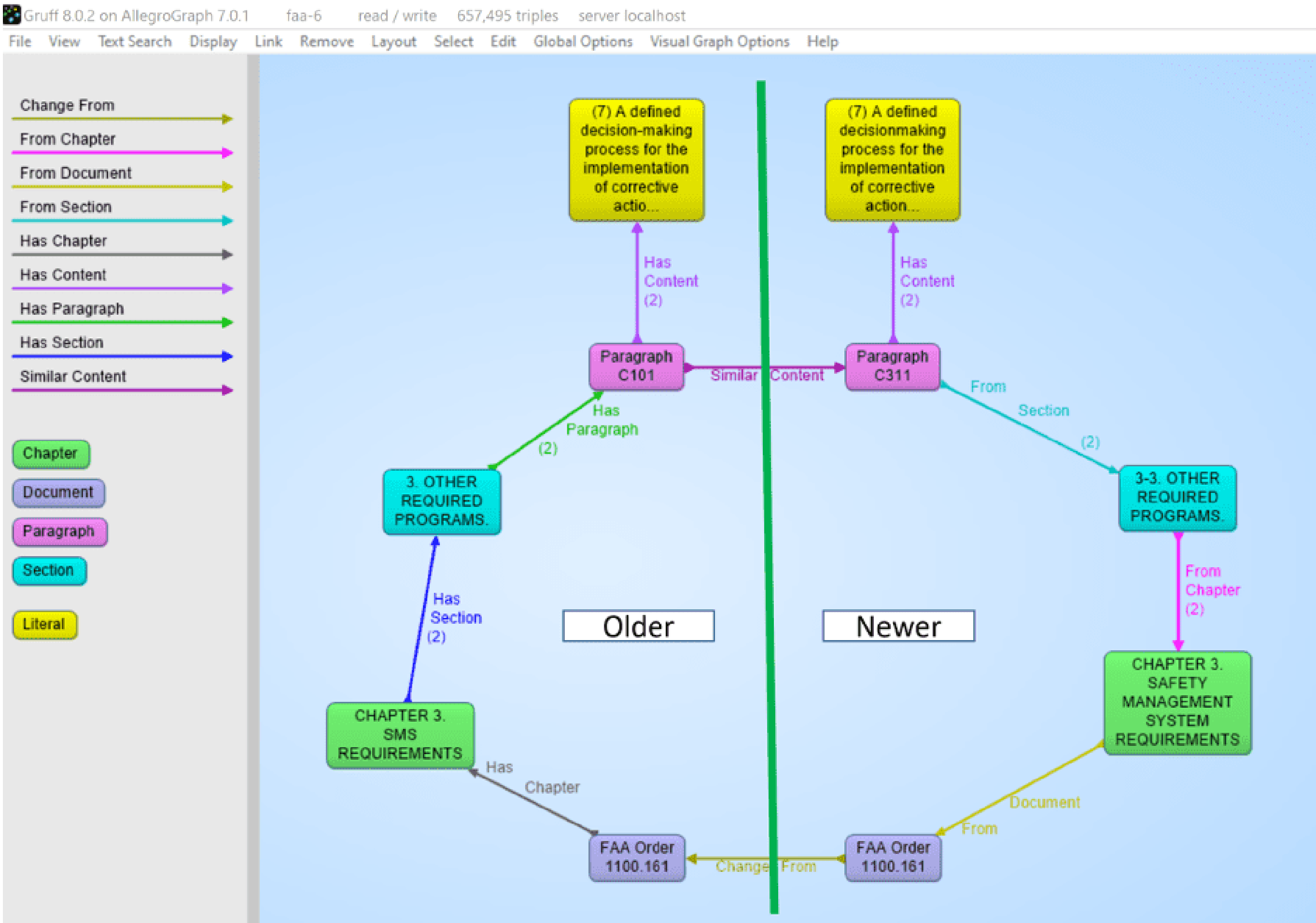
There are many many AI algorithms that can be applied in Document Knowledge Graphs. We provide best practices for topics like:

[a] Sentiment Analysis, using good/bad word lists or training data.
[b] Paragraph or Chapter similarity using statistical techniques like Gensim similarity or symbolic techniques where we just the overlap of recognized entities as a function of the size of a text.
[c] Query answering using word2vec or more advanced techniques like BERT
[d] Semantic search using the hierarchy in SKOS taxonomies.
[e] Summarization techniques for Abstractive or Extractive abstracts using Gensim or Spacy.

[9] Versioning and Document tracking

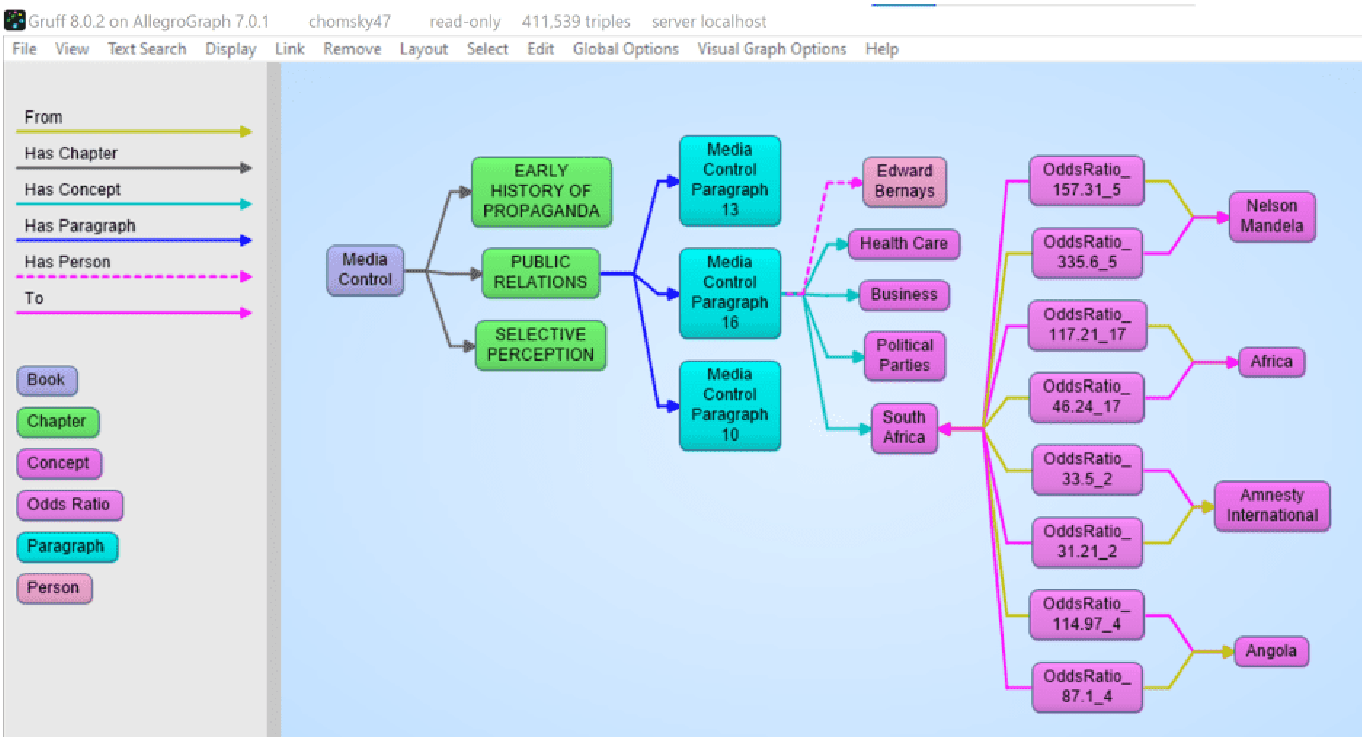
Several of our customers with Document Knowledge Graphs have noted the one constant in all of these KGs is that documents change over time. As part of our solution, we have created best practices where we deal with these changes. A crucial first step is to put each document in its own graph (i.e. the fourth element of every triple in the document tree is the document id itself). When we get a new version of a document the document ID changes but the new document will point back to the old version. We then compute which paragraphs stayed the same within a certain margin (there are always changes in whitespace) and we materialize what paragraphs disappeared in the new version and what new paragraphs appeared compared to the previous version. Part of the best practice is to put the old version of a document in a historical database that at all times can be federated with the ‘current’ set of documents.

Note that in the following picture we see the progression of a document. On the right hand side we have a newer version of a document 1100.161 with a chapter -> section -> paragraph -> contents where the content is almost the same as the one in the older version. But note that the newer one spells ‘decision making’ as one word whereas the older version said ‘decision-making’. Note that also the chapter titles and the section titles are almost the same but not entirely. Also, note that the new version has a back-pointer (changed-from) to the older version.



[10] Statistical Relationships

One important analytic one can do on documents is to look at the co-occurrence of terms. Although, given that certain words might occur more frequently in text, we have to correct the co-occurrence between words for the frequency of the two terms in a co-occurrence to get a better idea of the ‘surprisingness’ of a co-occurrence. The platform offers several techniques in Python and Lisp to compute these co-occurrences. Note that in the following picture we computed the odds ratios between recognized entities and so we see in the following gruff picture that if Noam Chomsky talks about South Africa then the chances are very high he will also talk about Nelson Mandela.



The Knowledge Graph Cookbook

Recipes for Knowledge Graphs that Work:

- Learn why and how to build knowledge graphs that help enterprises use data to innovate, create value and increase revenue. This practical manual is full of recipes and knowledge on the subject.
- Learn more about the variety of applications based on knowledge graphs.
- Learn how to build working knowledge graphs and which technologies to use.
- See how knowledge graphs can benefit different parts of your organization.

- Get ready for the next generation of enterprise data management tools.

Dr. Jans Aasman, CEO, Franz Inc. is interviewed in the Expert Opinion Section.

> **“KNOWLEDGE GRAPHS AREN’T WORTH THEIR NAME IF THEY DON’T ALSO LEARN AND BECOME SMARTER DAY BY DAY”** – Dr. Aasman

INTERVIEWS

The creation of knowledge graphs is interdisciplinary. Good chefs regularly visit other restaurants for inspiration. We have asked experts working in the field of knowledge graphs and semantic data modelling to comment on their experience in this area. They have worked with various stakeholders in different industries, so that you, dear reader, may further develop your understanding of the topic.

JANS AASMAN

FRANZ

Dr. Jans Aasman is CEO at Franz Inc., a leading provider of Knowledge Graph Technologies (AllegroGraph) and AI-based Enterprise solutions. Dr. Aasman is a noted speaker, author, and industry evangelist on all things graph.

“KNOWLEDGE GRAPHS AREN’T WORTH THEIR NAME IF THEY DON’T ALSO LEARN AND BECOME SMARTER DAY BY DAY”

Click here to get the book as free PDF or Kindle version.

Important Update to Franz Inc.'s Customers and Partners

COVID-19 is having a dramatic impact on people, communities, and businesses around the world. Our thoughts are with those who have been affected by the virus. Our team has spent considerable time preparing for the weeks ahead with a directed focus on our employees, customers, and partners.

Our number one priority is the health and safety of our employees and those we do business with around the globe. We have taken several measures to ensure everyone's well-being, including; restricting travel, implementing a global work-from-home policy, and included COVID-19 as top management agenda items for all meetings in order to closely monitor and rapidly respond to updates.

We have also taken a number of measures to ensure that the COVID-19 crisis does not impact the quality of your experience with Franz Inc. As part of our Business Continuity plan, our support team will ensure we meet our SLAs and continue to seamlessly deliver world-class customer support. We are also actively developing an exciting major update to AllegroGraph and we look forward to sharing details with you in the coming weeks..

As always, if you have any questions or concerns, please reach out to our global Support team via email at support@franz.com. If your questions are related to sales or general customer service please email info@franz.com.

On behalf of everyone at Franz Inc., thank you for trusting us with your business. We wish you and your families safety and good health.

Best Wishes,

Jans Aasman
CEO – Franz Inc.

Answering the Question Why: Explainable AI

The statistical branch of Artificial Intelligence has enamored organizations across industries, spurred an immense amount of capital dedicated to its technologies, and entranced numerous media outlets for the past couple of years. All of this attention, however, will ultimately prove unwarranted unless organizations, data scientists, and various vendors can answer one simple question: can they provide Explainable AI?

Although the ability to explain the results of Machine Learning models—and produce consistent results from them—has never been easy, a number of emergent techniques have recently appeared to open the proverbial 'black box' rendering these models so difficult to explain.

One of the most useful involves modeling real-world events with the adaptive schema of knowledge graphs and, via Machine Learning, gleaning whether they're related and how frequently they take place together.

When the knowledge graph environment becomes endowed with an additional temporal dimension that organizations can traverse forwards and backwards with dynamic visualizations, they can understand what actually triggered these events, how one affected others, and the critical aspect of causation necessary for Explainable AI.

Read the full article at AIthority.

Improving Data Processes with Knowledge Graphs

AllegroGraph Thought Leadership Article from Big Data Quarterly

Knowledge graphs link together data of any variety, structure, or format in business terms via uniform data models. Organizations can then join and traverse all of their data, semantically tagged with unique machine-readable identifiers, making the platform ideal for intelligent systems, machine learning analytics, interoperability, and an array of other benefits influential for AI applications.

The technology is gaining the attention of research firms and consultancies. In 2018 and 2019, knowledge graphs appeared on Gartner's Hype Cycle for Emerging Technologies, acknowledged for their hearty connections to pertinent data. According to Gartner, "These ecosystems developed as digitalization morphed traditional value chains, enabling more seamless, dynamic connections to a variety of agents and entities across geographies and industries. In the future these will include decentralized autonomous organizations (DAOs), which operate independently of humans and rely on smart contracts."

Download the Full White Paper.

100 Companies That Matter in Knowledge Management

Franz Inc., is proud to announce that it has been named to The 100 Companies That Matter in Knowledge Management by KMWorld. The annual list reflects the urgency felt among many organizations to provide a timely flow of targeted information. Among the more prominent initiatives is the use of AI and cognitive computing, as well as related capabilities such as machine learning, natural language processing, and text analytics.

"Knowledge management software and services providers are embracing a fresh wave of technological innovation to address heightened expectations—among both customers and employees—for the right information to be delivered to the right people at the right time, said Tom Hogan, Group Publisher at KMWorld. "To showcase organizations that are advancing their products and capabilities to meet changing requirements, KMWorld created the annual list of 100 Companies That Matter in Knowledge Management."

"We are honored to receive this acknowledgement for our efforts in delivering Enterprise Knowledge Graph Solutions," said Dr. Jans Aasman, CEO, Franz Inc. "In the past year, we have seen demand for Enterprise Knowledge Graphs take off across industries along with recognition from top technology analyst firms that Knowledge Graphs provide the critical foundation for artificial intelligence applications and predictive analytics. Our AllegroGraph Knowledge Graph Platform Solution offers a unique comprehensive approach for helping companies accelerate the creation of Enterprise Knowledge Graphs that deliver new value to their organization.”

How To Avoid Another AI Winter

Forbes published the following article by Dr. Jans Aasman, Franz Inc.’s CEO.


Photo: GETTY

Although there has been great progress in artificial intelligence (AI) over the past few years, many of us remember the AI winter in the 1990s, which resulted from overinflated promises by developers and unnaturally high expectations from end users. Now, industry insiders, such as Facebook head of AI Jerome Pesenti, are predicting that AI will soon hit another wall—this time due to the lack of semantic understanding.

“Deep learning and current AI, if you are really honest, has a lot of limitations,” said Pesenti. “We are very, very far from human intelligence, and there are some criticisms that are valid: It can propagate human biases, it’s not easy to explain, it doesn’t have common sense, it’s more on the level of pattern matching than robust semantic understanding.”

Read the full article at Forbes.

California utilities should have used digital twin technology instead of power shutoffs

Northern California's proactive power outages were not necessary last fall. Digital Twin technology can predict utility line failures and turn off power in milliseconds to avoid the potential of sparks igniting the surrounding area.

Digital twin technologies are gaining traction across industries and use cases. Initially devised as a means of monitoring assets and production settings in manufacturing, this technology has quietly seeped into other verticals like hospitality, construction, and building management and soon, electricity delivery.

The premier problem digital twins will solve is predicting power grid failure, which would alleviate the social, economic, and political issues that resulted from efforts to reduce the incidence and degree of catastrophes, property loss, and deaths stemming from downstream effects of power grid failure—such as recurring wildfires.

Digital twins can allay these concerns because they're based on real-time signals from a comprehensive set of factors that could be indicative of power grid woes related to environmental, meteorological, or technology concerns. Moreover, they can deliver accurate predictions for each of these factors well in advance of failure—in some cases as much as 28 days.

Read the full article at PowerGrid International.

Franz Inc. to Present at The Global Graph Summit and Data Day Texas

Dr. Jans Aasman, CEO, Franz Inc., will be presenting, “Creating Explainable AI with Rules” at the Global Graph Summit, a part of Data Day Texas. The abstract for Dr. Aasman’s presentation:

> *“There’s a fascinating dichotomy in artificial intelligence between statistics and rules, machine learning and expert systems. Newcomers to artificial intelligence (AI) regard machine learning as innately superior to brittle rules-based systems, while the history of this field reveals both rules and probabilistic learning are integral components of*

> *AI. This fact is perhaps nowhere truer than in establishing* explainable AI*, which is central to the long-term business value of AI front-office use cases.”*
>
> *“The fundamental necessity for explainable AI spans regulatory compliance, fairness, transparency, ethics and lack of bias — although this is not a complete list. For example, the effectiveness of counteracting financial crimes and increasing revenues from advanced machine learning predictions in financial services could be greatly enhanced by deploying more accurate deep learning models. But all of this would be arduous to explain to regulators. Translating those results into explainable rules is the basis for more widespread AI deployments producing a more meaningful impact on society.”*

The Global Graph Summit is an independently organized vendor-neutral conference, bringing leaders from every corner of the graph and linked-data community for sessions, workshops, and its well-known before and after parties. Originally launched in January 2011 as one of the first NoSQL / Big Data conferences, Data Day Texas each year highlights the latest tools, techniques, and projects in the data space, bringing speakers and attendees from around the world to enjoy the hospitality that is uniquely Austin. Since its inception, Data Day Texas has continually been the largest independent data-centric event held within 1000 miles of Texas.