Better Together: Extending Nestor with Technical Documentation

Jim Kukla
Co-Founder/CTO @ RedShred
Our journey to TLP

- Started with SBIR with USAF for processing unstructured technical orders into structured form
- Goals:
  - Accelerate troubleshooting
  - Improve mission-capable rates
  - Enable predictive maintenance
  - Extract structured information to integrate with databases
- Realized linking with MWO notes would be a separate challenge
  - Enter: Mike and Thurston
RedShred and the Ugly Doc-lings
RedShred: Your Documents as a Database

- **API-first platform**
  - Hybrid **vision and NLP** extraction and segmentation
    - Text, pictures, diagrams, tables
  - **Full-text** and **enrichment metadata search**
  - Easy to integrate into downstream applications

- **Collaboration**
  - Collection dashboard and **feedback APIs**
RedShred: Your Documents as a Database

- Configurable segmentation, extraction and enrichment

- Enrichments allow you to extract tailored metadata for your content (NLP, embeddings, figure analysis, etc.)

- Supports proprietary, open source, and third-party API enrichments

- Centralized resource to collaborate and maximize the value you get from your documents
RedShred + NIST in 2020

• RedShred provides programmatic access to technical documentation ...

• Nestor provides rapid tagging and annotation for MWOs ...

• 2020 NIST SBIR - Extended Nestor Tagging
  ○ The goal is to extend what Nestor does and tie it to large documents in RedShred
    ■ MWOs + Technical Documentation
Extending Nestor

1. Pluggable metrics
2. Concurrency
3. Custom Tags and Custom Classes
4. Integrated scoring feedback
5. Continuous on-the-fly deployment of tag models as an API
6. Integrated comparative visualizations
1. Pluggable Metrics

- Similarity
  - Edit-distance
  - Embeddings

- Ranking
  - TF-IDF
  - Weighting from business value columns

Embeddings from content fuel similarity and tags from Nestor become training data.

On-the-fly swapping to maximize coverage of synonyms.
2. Concurrency

- Shared tag model

- Independent tag models
3. Custom Tags and Custom Classes

- Arbitrary tags
- Item/Problem/Solution classification
- Ambiguous/Irrelevant classification

- Arbitrary tags
- User-defined multi-class attributes
  - Item/Problem/Solution classification
  - Ambiguous
  - Irrelevant
  - Department
  - Team
  - External link (database, catalog, etc.)
4. Integrated Scoring Feedback

- Allow users to **thumbs up/down individual examples in context**

- **Measures performance** of tag application
  - Recorded as **ground truth**

- From tags towards **training data** for named-entity recognition (NER) models
  - Facilitates links from MWOs to technical documentation
5. Continuous on-the-fly deployment

- Continuous on-the-fly deployment of a tag model as an API
  - ... and as an enrichment inside of your RedShred collection
  - Low-friction integration into point-of-data entry
6. Integrated Comparative Visualizations

- Small Multiples allow rapid comparison across metadata classes
  - Physical location (building, site)
  - Maintainer-specific tag models
    - Inter-annotator agreement for training and QA purposes
Better Together ❤

- **Extracted knowledge** from technical documentation
  - Embeddings, language models
  - Named entities and relationships
  - Text, figures, tables, images

- **MWO insights** from maintainers over time
  - Lived experience of fielded equipment
  - Personalized to each maintainer
    - ... with immediate, continuous deployment
  - Comparisons across maintainers
    - Collective wisdom across teams large and small
Conclusion

- Prototype in development, beta in late 2021
- Looking for research collaborators and beta testers - hello@redshred.com

Thank you!