THE UNIVERSITY OF MELBOURNE

HUNT LAB FOR INTELLIGENCE RESEARCH

School of BioSciences
Parkville, Victoria, Australia

huntlab.science.unimelb.edu.au

Enhancing human analysis of large corpora via argument mining and backlinks—a prototype

March 2021

Researchers

Luke Thorburn   luke.thorburn@unimelb.edu.au

Contact

A/Prof. Tim van Gelder

Director
Hunt Lab for Intelligence Research
University of Melbourne

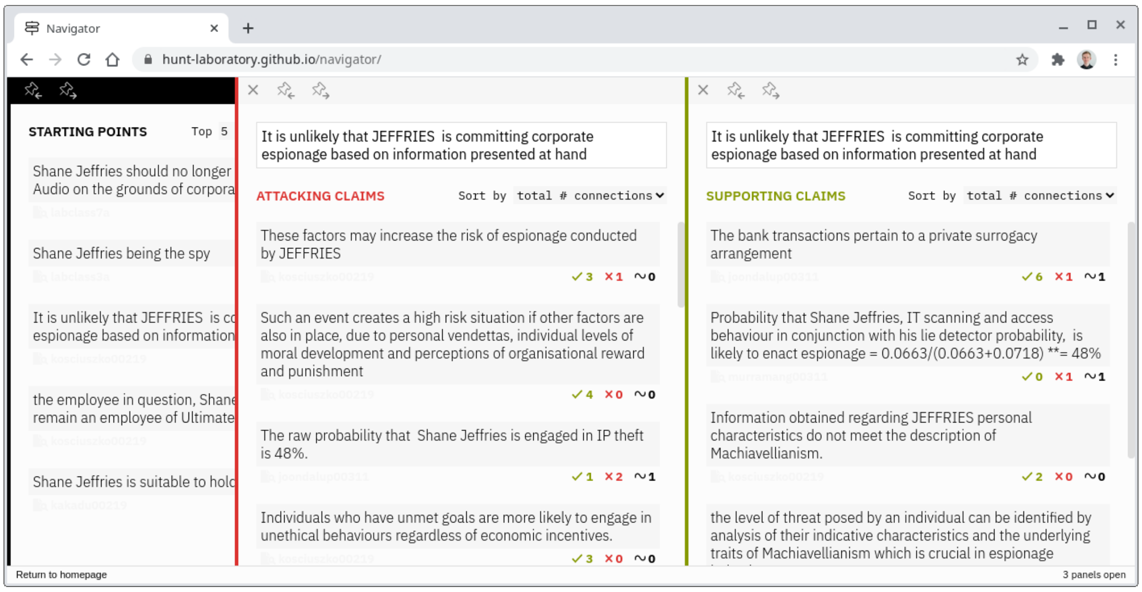
t.gelder@unimelb.edu.au
+61 438 131 266

1 Non-Technical Overview

Information overload is pervasive in intelligence analysis [23, p. 22]. This technical report documents the development of *Navigator*—an experimental, prototype application designed to enhance an analyst's ability to quickly extract insight from a large corpus of documents related to a given topic. It is designed to help answer questions such as:

1. What claims are being made, and with what frequency?
2. Who is making them?
3. What arguments are presented for or against each claim?
4. Which claims are most/least contested?

Beyond augmenting the abilities of individual analysts, the tool addresses a fundamental challenge facing the establishment of a competition-based marketplace for citizen intelligence work—namely, the ability to extract insight from submitted intelligence reports. If successfully implemented, such a marketplace would increase intelligence organisations' access to high-quality analysis, whilst improving public trust [8].



STARTING POINTS Top 5

Shane Jeffries should no longer
Audio on the grounds of corpora

labclass7a

Shane Jeffries being the spy

labclass3a

It is unlikely that JEFFRIES is co
espionage based on information

kosciuszko00219

the employee in question, Shane
remain an employee of Ultimate

kosciuszko00219

Shane Jeffries is suitable to hold

kakadu00219

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand

ATTACKING CLAIMS Sort by total # connections

These factors may increase the risk of espionage conducted by JEFFRIES
kosciuszko00219 ✓3 ✗1 ∼0

Such an event creates a high risk situation if other factors are also in place, due to personal vendettas, individual levels of moral development and perceptions of organisational reward and punishment
kosciuszko00219 ✓4 ✗0 ∼0

The raw probability that Shane Jeffries is engaged in IP theft is 48%.
joondalup00311 ✓1 ✗2 ∼1

Individuals who have unmet goals are more likely to engage in unethical behaviours regardless of economic incentives.
kosciuszko00219 ✓3 ✗0 ∼0

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand

SUPPORTING CLAIMS Sort by total # connections

The bank transactions pertain to a private surrogacy arrangement
joondalup00311 ✓6 ✗1 ∼1

Probability that Shane Jeffries, IT scanning and access behaviour in conjunction with his lie detector probability, is likely to enact espionage = 0.0663/(0.0663+0.0718) **= 48%
murramang00311 ✓0 ✗1 ∼1

Information obtained regarding JEFFRIES personal characteristics do not meet the description of Machiavellianism.
kosciuszko00219 ✓2 ✗0 ∼0

the level of threat posed by an individual can be identified by analysis of their indicative characteristics and the underlying traits of Machiavellianism which is crucial in espionage

Return to homepage

3 panels open

Figure 1: A screenshot from Navigator, the prototype application.

The application has two components: an argument mining system and a graphical user interface.

Argument Mining

Argument mining is the automated extraction of reasoning structure from text. An argument mining system takes sets of documents (e.g., intelligence products, articles, transcripts or web pages) as input, and outputs a set of the claims made therein, along with the argumentative relations between them. Argument mining is a rapidly-developing area of artificial intelligence, in the sub-field of natural language processing.

In Navigator, users are able to upload their own texts. These are then analysed by an argument mining system that uses two neural networks to first identify claims in the source texts, and then classify the argumentative relations between the identified claims. This abstract argument graph is then presented to the user via a graphical user interface (see below).

Performance of the argument mining models is sufficient to serve as a proof of concept, but is necessarily imperfect. This is partly due to the constraints of the project and partly due to the fact that this work tackles challenges that have received little previous research attention, such as the fact that we are performing multiple-document (rather than single-document) argument mining.

Improving the reliability of the tool will likely require the creation of new datasets tailored to the intelligence context, as well as innovation in the design of neural networks for argument mining.

Further details of the argument mining system can be found in Section 3.

User Interface

Most commonly, argumentative structure is visualised as graph, where nodes correspond to claims and edges correspond to relationships between claims. Such user interfaces can be cumbersome to interact with on screens, and anecdotal evidence suggests that they struggle to attract widespread adoption.

Navigator has a novel user interface (Figure 1) that facilitates exploration of the extracted arguments, without making the graph structure explicit. It is modelled on recent innovations in the design of notetaking applications such as Roam Research and Obsidian. These applications have bidirectional links or backlinks, which make it easy to see which notes link *to* any given note. (Picture a situation in which at the bottom of every webpage there was a list of all the pages that link to that page.) Navigator generalises the concept of bidirectional links to capture the different types of relations (support, attack, equivalence) present in argument maps.

The interface is already quite usable, but could benefit from improvements to allow it to save and restore the currently-open set of panes, to better preserve the context in which panes were opened, and to visually aggregate claims that are sufficiently similar.

Further details of the user interface can be found in Section 4.

The prototype can be accessed at `https://hunt-laboratory.github.io/navigator/`.

2 Technical Overview

Navigator is an experimental, prototype application designed to enhance an intelligence analyst's ability to quickly extract insight from a large corpus of documents. It was developed as part of the AI for Decision Making Program, a collaborative initiative of a number of Australian defence science organisations.

This report separately documents the two main components of the application: an argument mining system and a graphical user interface.

Argument Mining

A pipeline of two neural networks extracts a multi-document argument graph from a set of user-uploaded texts. The two neural networks respectively perform the argument mining sub-tasks of argument component identification and argument relation classification.

Argument component identification is structured as an IOB sequence tagging problem over token embeddings, and performed using a bidirectional LSTM. Argument relation classification is performed using a network of two identical bidirectional GRUs, augmented with a multiplicative attention mechanism.

Given the constraints of the project, performance of the models is below state-of-the-art, but sufficient to serve as a proof of concept. New dataset creation and a transition to an end-to-end argument mining system are likely necessary directions for future work, in order to improve the reliability of the tool.

Full details of the datasets, model architectures, deployment infrastructure, evaluation, and anticipated directions for future work can be found Section 3.

User Interface

The user interface provides a novel way of interacting with a multi-document argument map for the purposes of surfacing potentially relevant evidence and hypotheses.

It is modelled on recent innovations in the design of notetaking applications such as Roam Research and Obsidian. These applications can be viewed more generally as interfaces for interacting with graph structures over sections of text, one example of which is graphs of argumentative structure. Drawing on this analogy, Navigator generalises the concept of bidirectional links or backlinks used in these applications to capture the typed relations (support, attack, semantic similarity) present in argument maps.

The interface is already quite usable, but could benefit from improvements to allow it to save state, preserve context and aggregate semantically similar claims, as well as to undergo usability studies.

A full rationale for the user interface design, a review of relevant history of argument visualisation and networked note repositories, technical details of the interface implementation and anticipated directions for future work can be found in Section 4.

The prototype can be accessed at `https://hunt-laboratory.github.io/navigator/`.

Contents

3 Argument Mining

3.1 Background

Argument mining—the automatic extraction of the argumentative structure of prose or dialogue—is a natural language processing (NLP) task that is gaining increasing attention, though it remains more niche than many other NLP tasks due its relative complexity and a shortage of high-quality labelled datasets. Interpretation of the task itself is also inconsistent, with many variants occuring in the literature [41]. Broadly speaking, the goal is to input a text (be it a scientific article, opinion piece, forum thread, chat history, or something else) and return an argument map. An example of an argument map can be seen in Figure 6. In contrast to manually-created argument maps, the nodes in an argument map returned from an argument mining algorithm are generally verbatim or lightly processed excerpts from the original text, and may not be standalone, well-phrased statements when viewed in isolation. For recent surveys of the field, see [18, 15].

Argument mining is commonly decomposed into a pipeline of simpler tasks. In this project, we used two steps.

1. **Segmentation** First, the text is segmented into spans that each correspond to at most one argumentative unit: a proposition or statement that could serve as a premise, reason or conclusion to an argument. For brevity, we will refer to such statements as *claims* (acknowledging that in the literature, the term 'claim' often refers specifically to the conclusion of an argument).

2. **Classification** Next, pairs of claims are classified on the basis of the argumentative relationship between them. For example, given claims a and b, does a support b? Does b attack a? Do a and b mean the same thing? Or are they unrelated?

Segmentation is usually approached as a sequence-tagging task, often using the IOB convention [27]. In this setup, the input text is tokenized, and then each token is assigned a label: `B` if it marks the beginning of a claim, `I` if it falls elsewhere within a claim, and `O` if it is not part of any claim. In effect, this is a three-class classification problem for each token in the text.

Inspired by the state-of-the-art performance of a BiLSTM-CNN-CRF architecture on sequence tagging tasks such as named entity recognition and part-of-speech tagging [19], a number of recent papers have used this framework for argumentative unit extraction [5, 37]. In this approach, the BiLSTM is the base model, with a convolutional neural net (CNN) used to produce character-level embeddings to describe words not seen during training, and a Conditional Random Field (CRF) [13], which is a purely probabilistic model, used to capture dependencies between adjacent tags (such as the fact that an `I` can never directly follow an `O`).

This combined approach supercedes previous attempts that used CRFs directly, working with extracted properties of the input text [35, 32]. This approach was found to struggle to capture long-range dependencies between tokens. Exclusively deep learning approaches have also been used, such as the layering of multiple Bi-LSTMs described in [1].

Yet earlier attempts simply classify whole sentences as argument or non-argument based on extracted properties [21], or used two Naive Bayes classifiers based on properties of local 3-grams, to classify words as demarcating the beginning or end of a claim [17].

The second task, argument relation classification, has also seen a number of approaches employed. Lawrence and Reed [16] explored how discourse indicators ('because', 'therefore', 'however') may be used to identify argumentative relations, and conclude that their use can achieve high precision but low recall, because only a small proportion of relations between claims are signposted in this way.

Recent work relies on recurrent neural networks over token embeddings to produce representations of two given claims, which are then fed into a classifier. Cocarascu et al. [6] explore four different architectural variants of this approach (which variously make use of GRUs, autoencoders and attention mechanisms), and assess their performance on a diverse set of argument-annotated datasets. They find that with non-contextual embeddings, an attention-based model tends to perform best across multiple datasets.

A fundamental limitation of pipeline approaches to argument mining is that, because each subtask is completed imperfectly, error will accumulate as the data proceeds through the sequence of tasks. One alternative is to train all the models simultaneously, and jointly optimise the parameters of each model. Equivalently, the task can be structured such that it can be completed in a single step, by a single model. Both approaches fall under the umbrella of *end-to-end* argument mining. Though this is the not the approach taken here, we believe that end-to-end argument mining systems will emerge to consistently outperform pipelines with independently-trained components, and that this should be the approach taken going forward. See Section 3.6 for more on end-to-end argument mining.

The needs of this project went beyond existing literature in a number of directions. To our knowledge, the following aspects of the problem have received little prior attention.

- **Multiple texts** The system needs to be able to extract a combined argument map from a potentially large number of documents. Existing literature is primarily concerned with single-document argument mining.
- **Multiple text types** To be useful to analysts, the system may need to be sufficiently general that it could work with multiple text types. Most of the existing models used for argument mining are trained and evaluated on a single, stylistically homogeneous dataset. Previous work has considered multiple topics [34], but not multiple text types.
- **Relationship types** Most existing systems for argument relation classification assume that there are at most three types of relationships: *support*, *attack*, or *no relation*. To be useful, particularly when mining arguments from large corpora, we think it is a requirement to consider a fourth relationship: whether two claims are *semantically similar*. If detected reliably, such a relationship could be used to 'collapse' nodes in the resulting argument graph, which would simplify the structure, reduce redundancy and help identify support or attack relations that may otherwise not have been found. Though similarity or argumentative rephrase relationships have been considered in the literature [12, 30, 2], they are rarely included as a class in argument relation classifiers.

3.2 Datasets

Due to the limited time available for this project, it was necessary to make use of existing argument-annotated corpora. In order to explore the feasibility of training a single system that would work across multiple domains and text-types, we aggregated multiple datasets: three for the segmentation task (S), and five for the classification task (C). The following table describes each dataset, the processing that was required to convert it to a common format, and how each dataset was partitioned into training, validation and test sets.

Dataset	Tasks	Citation
IBM Debator Claims and Evidence 547 Wikipedia articles of relevance to one of a set of 58 debate topics, such as 'the sale of violent video games to minors should be banned'. Spans in each article are highlighted if they represented a claim about the topic, or a piece of evidence that supports a claim. In total, the dataset contains 2,294 claims, and 4,692 evidence phrases. This dataset was structured such that the claim and evidence phrases were stored in spreadsheets, separated from the original article text. Thus in order to format the data for an IOB sequence tagging task, the claim and evidence phrases needed to be located and annotated in the original text. This was performed with the aid of the `PhraseMatcher` function within the spaCy Python library. The matching was imperfect, due to differences in the phrases between the spreadsheets and	S + C	Rinott et al. [31]

Dataset	Tasks	Citation
the original texts. For example, sometimes new line characters that occurred mid-phrase in the original text were not included in the spreadsheet. Similarly, in some cases the tokenisation of the reported phrase did not match up with the tokenisation of that phrase in the original article. Reasonable efforts were made to overcome these challenges, including searching for slightly truncated versions of the phrases to avoid tokenisation issues at the phrase boundaries. Ultimately, 112% of the claims were re-identified (this is possible due to the truncation), and 101% of the evidence phrases. Articles with fewer than 5 annotated phrases were removed, reducing the number of articles to 255 for the segmentation task. All relevant pairs of evidences and claims from the full dataset were added to the classification dataset as examples of support relations. Additionally, 1,000 unrelated pairs of statements were randomly sampled from different topics (to give a high chance of being truly unrelated). In the original dataset, debate topics were already assigned to either a 'train and test' or 'held-out' tranche. We used the 'train and test' set as our training set, and randomly assigned half of the 'held-out' set to be our validation set, and the other half to be our test set.		
Dr Inventor 40 scientific articles from the field of computer graphics, annotated for three types of claims (`own_claims`, `background_claim` and `data`) as well as three types of relations (`supports`, `contradicts`, `semantically_same`). The data was formatted as text files of the original articles, along with TSV spreadsheets describing the annotations for each article, in the format used by the Brat annotation tool. Thus, as with the first dataset, the extracted phrases needed to be relocated in the original articles for sequence tagging. Each extracted phrase was provided with the character position of its start and end points within the article, so these were used for reidentification. Due to the fact that some of the character indices did not align with token boundaries, only 99.12% of phrases were successfully reidentified. All argument relations from the original dataset were added to the classification dataset with the labels more-or-less preserved. Pairs annotated as `semantically_same` were relabelled as `semantically_similar`, because we wanted to broaden the strictness of that class to be compatable with other datasets used. Additionally, for each article 50 pairs of claims with no relation specified were randomly sampled and added as examples of the `no_relation` class. This dataset was not already partitioned, so we randomly assigned 30 articles to our training set, and 5 each to the validation and test sets.	S + C	Lauscher, Glavaš, and Ponzetto [14]

Dataset	Tasks	Citation
Persuasive Essays 402 short persuasive essays on almost as many topics. The data was formatted identically to that of the Dr Inventor corpus, and so the phrase re-identification process was analogous. 99.92% of phrases were successfully re-identified. The extraction of claim pairs for classification is also analogous to the previous dataset. Because the essays were shorter than the scientific articles, in each essay 20 randomly-selected pairs of claims with no relation specified were added as examples of the `no_relation` class. This dataset was already partitioned into 'train' and 'test' sets. We left the test set as is, but further randomly selected the same number of essays as was in the test set from the 'train' set to be our validation set. The remainder were assigned to the training set.	S + C	Stab and Gurevych [33]
IBM Debator Claim Stance Classification 2,394 claims extracted from Wikipedia articles related to one of 55 debate topics. Each is annotated with `PRO` or `CON`, specifying the relationship of the claim to the topic. This dataset was only used for the relation classification task, so the claims did not need to be relocated in the original articles. `PRO` labels were mapped to `support`, and `CON` labels were mapped to `attack`. Topics were assigned to a 'train' or 'test' partition. We left the training set as is and randomly split the test partition, with half of those topics become our validation set. The remainder were left as the test set.	C	Bar-Haim et al. [3]
UKP ASPECT Argument Similarity 3,595 pairs of sentences, each labelled with the degree of similarity between them. The labels were 'Different Topic/Can't decide' (`DTORCD`), 'No Similarity' (`NS`), 'Some Similarity' (`SS`) and 'High Similarity' (`HS`). The labelling captured whether the two sentences belong to the same topic, and if so, whether they are talking about the same aspect of that topic, or are saying the same thing using different words. This dataset was only used for the relation classification task, so the claims did not need to be relocated within their origin texts. All relation labels were mapped to `no_relation`, except for `HS` which was mapped to `semantically_similar`. Of the 28 topics, 18 were randomly assigned to the training set, and 5 each to the validation and test sets.	C	Reimers et al. [30]

As we intended for our classification model to work in both directions (that is, to be equally informative regardless of the order in which the two claims are inputted), all pairs of claims were added to the dataset in both directions. The set of classes used in the classification dataset is as follows.

- `no_relation`
- `supports`

- `is_supported_by`
- `attacks`
- `is_attacked_by`
- `semantically_similar`

After the processing described above, we had created one dataset each for the segmentation and classification tasks.

The segmentation dataset contained 442 documents and 19,557 claims. The classification dataset contained 64,170 pairs of claims. This is comprised on 28,350 `no_relation`, 15,421 `supports`, 15,421 `is_supported_by`, 1,985 `attacks`, 1,985 `is_attacked_by`, and 1,008 `semantically_similar`.

Originally, we had intended to include in the training data a corpus of intelligence products written during a series of exercises on the Hunt Lab SWARM platform, most notably the 2018 SWARM Challenge and the 2020 Hunt Challenge. These reports are authored by teams of analysts, who were either professional analysts of organisations with intelligence capabilities, members of the public recruited from the Hunt Lab's research pool, or university students. In the end, full annotation of these reports proved infeasible in the time available, so they were not used for model training. We did, however, annotat 9 such reports, all addressing the same fictional problem which consisted of assessing the likelihood that a suspected employee of a high-end audio equipment company was conducting corporate espionage. Annotation was undertaken using the OVA annotation tool [10].

Though not used in for training the models, we believe that this type of dataset—where a number of documents are all related to a common topic—is one of the more promising contexts in which to apply this technology. As such, the dataset is available in the prototype application to explore for demonstration purposes.

While we originally intended to perform considerable hyperparameter tuning, this was not ultimately possible due to time constraints, so there was no real need to have both a validation and a test set. Thus, the test set went unused, but is available for future work.

3.3 Segmentation

3.3.1 Task

We structured the task as an IOB sequence tagging task. The model took in a sequence of token embeddings, and returned a softmaxed probability distribution over the 3 classes (`B`, `I`, `O`) for each token.

The token embeddings used were those from the pre-trained spaCy `en_core_web_md` language parser. Each vector had length 300. No features other than the semantic token vectors were used, and the embeddings were not finetuned on our dataset.

Loss Function Mean, unweighted cross-entropy loss over each token-level classification.

Evaluation Metric For the evaluation metric, we used a generalised version of the F_1 score designed to capture meaningful information about the success of segmentation tasks such as this. The F_1 $(x\%)$ score is calculated analogously to a conventional F_1 score, but with a different definition of a 'match'. Here, a predicted span is said to match a pre-labelled span if they share $x\%$ of their tokens [24]. The denominator of the percentage is taken to be the length of the longest of the two spans. Also, because our model must learn rules about what sequences of labels are valid, we manually correct `OI` to `OB` to allow the F_1 scores to be computed.

3.3.2 Architecture

Based on the recent successes of Bi-LSTMs for argument unit segmentation, we used a relatively simple architecture consisting of a single-layered Bi-LSTM, followed by a few fully-connected layers to produce a softmax classification for each token from the output of the Bi-LSTM. A schematic of the architecture used can be seen in Figure 2.

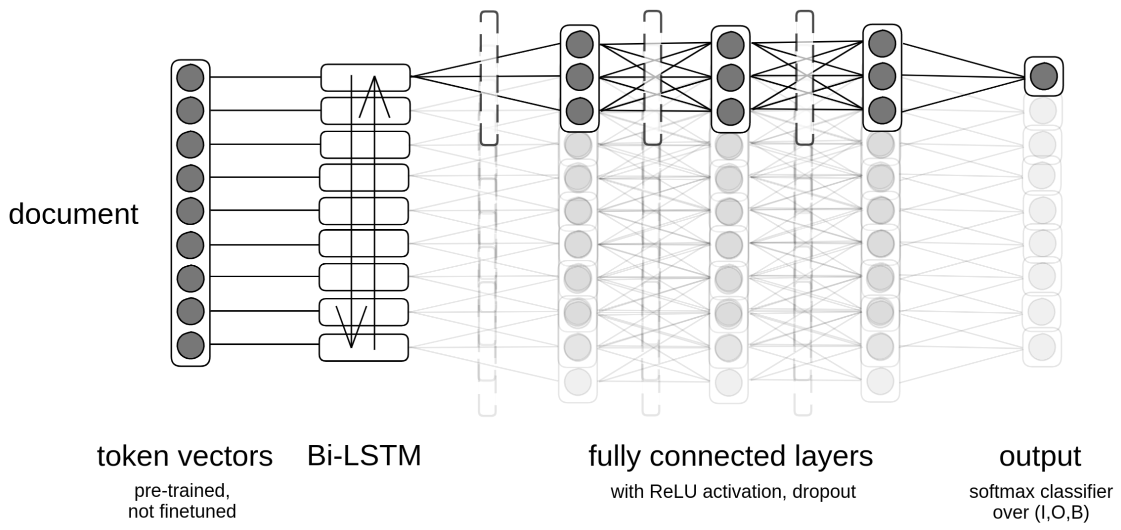

document
token vectors
pre-trained, not finetuned
Bi-LSTM
fully connected layers
with ReLU activation, dropout
output
softmax classifier over (I,O,B)

Figure 2: High-level architecture of the model used for the segmentation task.

Both the LSTM hidden state and the fully-connected classification layers were of size 100. The dropout probability used was 0.5.

3.3.3 Results

We trained the network using stochastic gradient descent with a mini-batch size of 16 for 370 epochs, which took about 20 hours on our cloud machine (8 Intel Xeon vCPUs, 1 NVIDIA Quadro M4000/P4000 GPU, 8GB RAM). The learning rate and momentum parameters were fixed at 0.1 and 0.9, respectively. The loss and evaluation metric values on the training and validation sets during training can be seen in Figure 3.

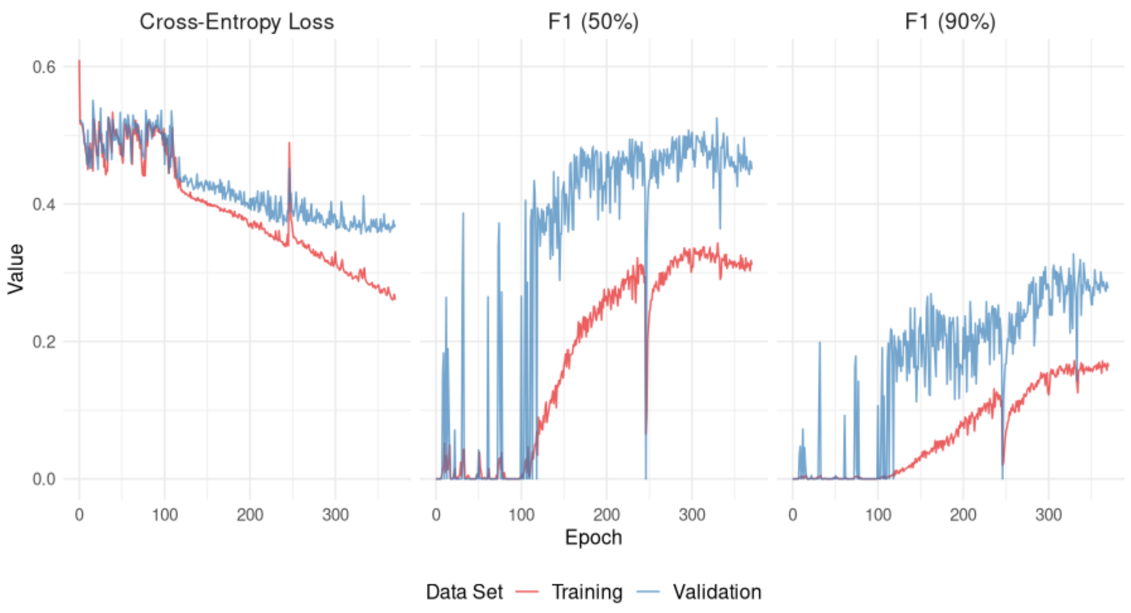


Figure 3: Loss and evaluation metrics during training for our segmentation model. It took about 100 epochs for the model to start learning. After 370 epochs, we decided to halt training as it appeared the model was starting to overfit. For deployment, we used the 305-epoch model.

The training was monitored throughout and was halted when it appeared the model was starting to overfit, as indicated by the plateauing of the validation cross-entropy loss and the deterioration in the validation F_1 (50%) score.

The model used in deployment was that which was saved at the 305-epoch checkpoint. For this model, the validation cross-entropy loss was 0.36, the F_1 (50%) score was 0.50, and the F_1 (90%) score was 0.31.

For context, state-of-the-art results on a comparable task obtained by Eger, Daxenberger, and Gurevych [7] produced an F_1 (50%) score of 0.51, and an F_1 (100%) score of 0.45. But that result was attained on a less diverse dataset (the persuasive essay dataset which is a subset of ours), and employed a more sophisticated model as well as hyperparameter optimisation, which we did not attempt due to time constraints.

In practice, for any text there is a large set of possible segmentations which could be considered admissible, of which the annotations in the training data are just a single example. Thus, while the F_1 scores obtained are low, in practice most of the claims extracted appear to be coherent.

3.4 Relation classification

3.4.1 Task

The task is a simple classification problem for each pair of claims. The model takes in two sequences of of token embeddings (one sequence for each of the two claims), and returns a softmaxed probability distribution over the six classes (`no_relation`, `supports`, `is_supported_by`, `attacks`, `is_attacked_by`, `semantically_similar`).

We chose to include symmetric pairs of classes (such as `supports` and `is_supported_by`) where applicable, so that the model would be able to classify relationships between the two claims in either direction.

Again, the token embeddings used were those from the pre-trained spaCy `en_core_web_md` language parser. Each vector had length 300. No features other than the semantic token vectors were used, and the embeddings were not finetuned on our dataset.

Loss Function Cross-entropy loss for each classification problem. Each of the classes were weighted inverse-proportionally to the number of examples in the dataset. This was to ensure the model was incentivised to learn to recognise classes with few examples.

Evaluation Metric For the evaluation metric, we used the conventional F_1 score, treating each of the classes as a binary classification problem.

3.4.2 Architecture

The architecture used is a modified version of the attention-based architecture which was found to be most successful across domains by Cocarascu et al. [6].

First, the sequence of token vectors for each claim are each fed through a bidirectional Gated Recurrence Unit (GRU) of hidden size 128 to produce a representation of each claim of size $(256, \texttt{n\_tokens})$.

Next, this representation of each claim is attended to the representation of the other, and these attended representations are concatenated to the representations of the respective original claims. The attention scores are computed using the same multiplicative attention formulae provided in the original paper [6, Sec. 3.2.4].

These contextual representations of each claim (of size $(256, 2 \times \texttt{n\_tokens})$) are then fed through a second bidirectional GRU. The final hidden states of these GRUs are concatenated row-wise into a vector of size $(1, 512)$ and fed through a single fully connected layer of size 128, and then to a softmax layer of size 6 (one element for each class). Either side of the fully connected layer, we apply dropout with probability 0.5, and follow it with ReLU activation.

Key differences between this architecture and that specified in [6] are that we use bidirectional (rather than unidirectional) GRUs, and that we do not make use of additional features that describe textual entailment, sentiment, or syntax.

A schematic of the architecture used can be found in Figure 4.

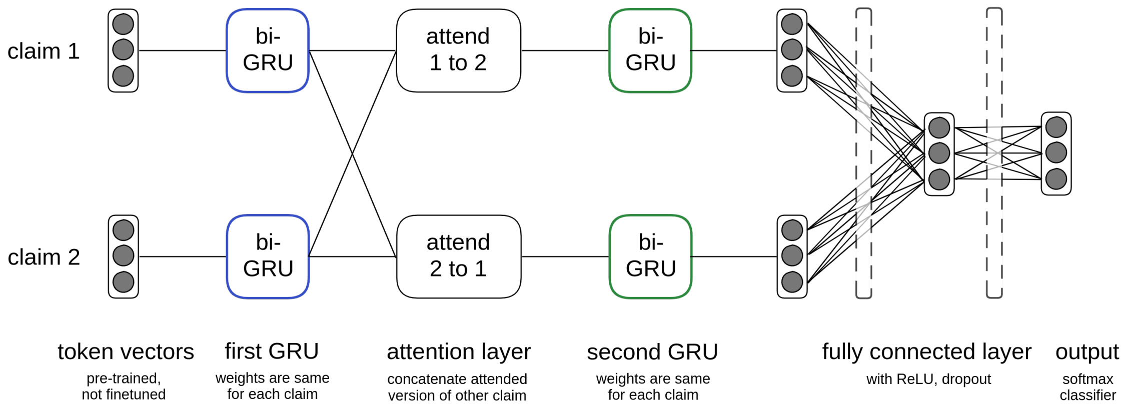


Figure 4: High-level architecture of the model used for the segmentation task.

3.4.3 Results

We trained the network using stochastic gradient descent with a mini-batch size of 64 for 210 epochs, which took about 8 hours on our cloud machine (8 Intel Xeon vCPUs, 1 NVIDIA Quadro M4000/P4000 GPU, 8GB RAM). The learning rate and momentum parameters were fixed at 0.1 and 0.9, respectively. The loss and evaluation metric values on the training and validation sets during training can be seen in Figure 5.

The training was monitored throughout and was halted when it appeared the model was no longer meaningfully learning. In most cases, the validation loss and evaluation metrics peaked at around the 20-epoch model, then deteriorated slightly and plateaued. The cross-entropy loss indicated severe overfitting after 210 epochs, but unusually this was not reflected in the F_1 scores for each class.

The model used in deployment was that which was saved at the 20-epoch checkpoint. For this model, the validation cross-entropy loss was 1.63, the F_1 scores for each class are given in the following table. Classes with a larger number of training examples were more successfully learned.

Class	F_1 Score
`no_relation`	0.621
`supports`	0.543
`is_supported_by`	0.541
`attacks`	0.163
`is_attacked_by`	0.165
`semantically_similar`	0.069

For context, Cocarascu et al. [6] obtained average F_1 scores of 0.449 for attack relations, and 0.640 for support relations. Our results fall short of this benchmark, though our task was likely made more challenging by the fact we trained on a pool of multiple datasets (rather than each dataset separately), required classification over a greater number of classes, and only used semantic features. We also relied on their hyperparameter values and did not perform our own optimisation due to time constraints.

Qualitatively, the performance of this model appears to be very hit-and-miss. Many of the classified argumentative relations are nonsense, particularly for those classes of which there were relatively few training examples. But successful classifications are not infrequent, and suggest that such models might obtain a useful degree of accuracy with sufficient time, data, and exploration of different architectures and hyperparameters.

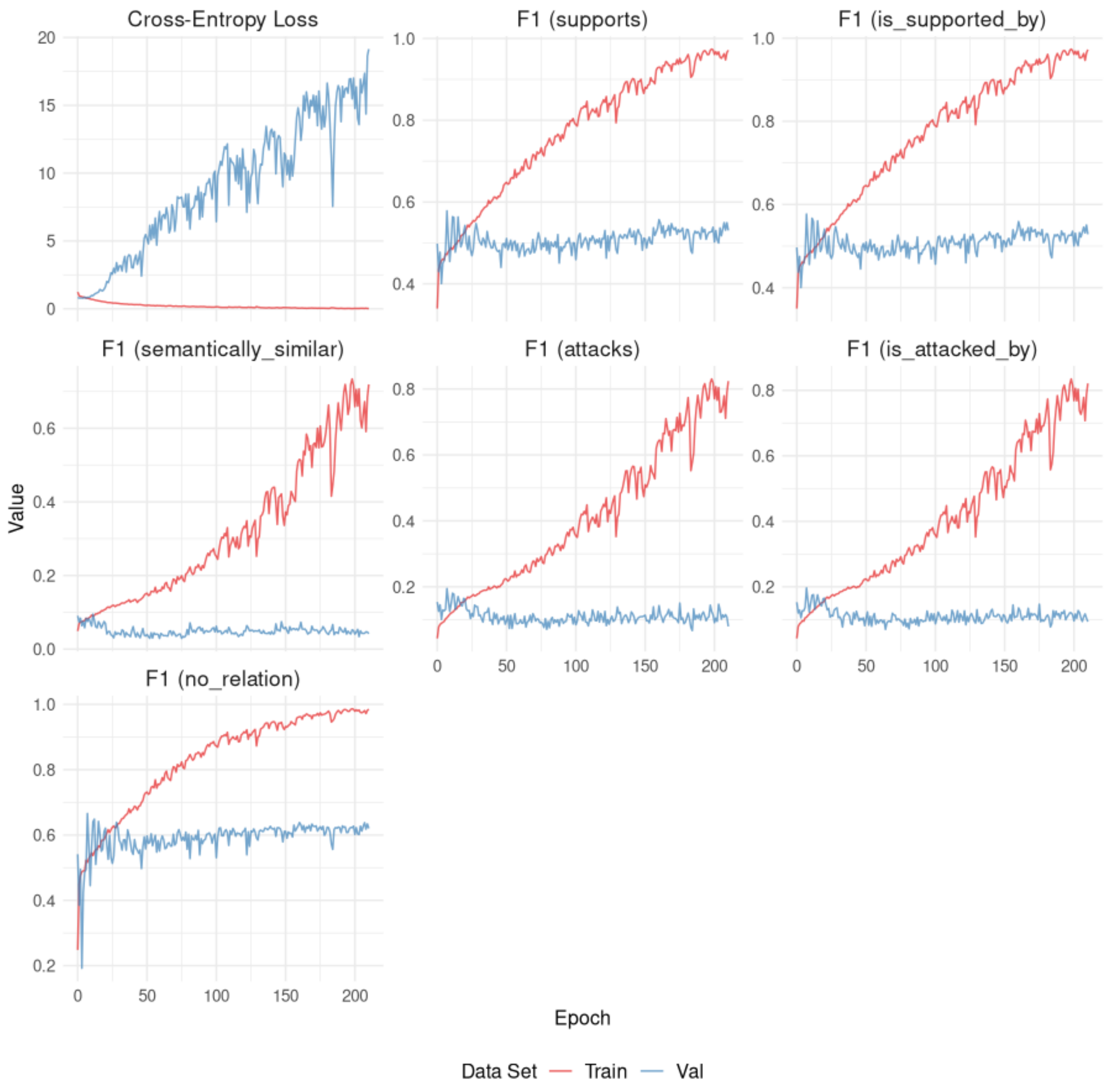


Figure 5: Loss and evaluation metrics during training for our relation classification model. Evaluation metrics plateaued fairly quickly as training progressed. In deployment, we used the 20-epoch model.

3.5 Deployment

We used Cortex to deploy the trained models to an Amazon Web Services cluster and set up an API to allow the models to be queried by the user interface (UI).

The AWS cluster used was in the `ap-southeast-2` region, was of instance type `m5.large`, and was attached to a 50GB `gp2` volume. This instance costs about USD0.33/hour to run.

With this available compute, the API can process up to around 12 medium length documents per call, before it times out after 30 seconds.

The API was structured such that it performs all required analysis of a set of documents in a single API call. At a high level, this involves:

1. Tokenising and vectorising the documents using the spaCy parser.
2. Passing the documents through the segmentation model to identify claims.
3. Producing candidiate pairs of claims from within each document that have a reasonable chance of being related. This is necessarily a subset of all possible relationships, because it it

computationally infeasible to consider all $\binom{\#\ \text{claims}}{2}$ possible relations. In our implementation, we considered possible relations between claims that were no more than 5 claims distant from one another.

4. Classifying all candidate within-document relations using the classification model.
5. Computing the Katz centrality of the claims within each document. Katz centrality is a graph theoretic measure of centrality that assigns a score to each vertex (in our case, each claim) based on the number and length of incoming paths (in our case, chains of reasoning) [11].
6. Generating candidate pairs of claims from *different* documents that have a reasonable chance of being related. There are many possible ways of doing this. In our case, we took the three most central claims from each document as measured by Katz centrality, and considered all possible cross-document pairings of this reduced set of claims.
7. Classifying all candidate cross-document relations using the classification model.
8. Formatting all annotations appropriately, including versions of each of the original texts marked up with HTML spans indicating the identified claims.

To allow analysis of a greater number of documents (which would likely be required in practice), the API calls could be restructured such that each model query requires its own API call, and all the peripheral, non-ML data wrangling could be relocated to the client side. This would also have the added benefit of being able to display progress messages to the user as the analysis progresses.

However, if we shift to an end-to-end argument mining approach, as is proposed in the following section, then most of the analysis would necessarily need to take place within a single API call. In this case, increasing the capacity of the system would require optimising the server-side analysis code, increasing the available compute, and/or increasing the timeout threshold on the API.

3.6 Future work

This section describes directions for future work, many of which would be required for this technology to mature to the point where it is robust enough to be useful for routine use by analysts.

Dataset creation We are not aware of any labelled datasets currently available for training models to perform multi-document argument mining. At best, we can do what we have done here and use datasets for single-document argument mining, under the assumption that inter-document argumentative relations will be of a similar nature to intra-document argumentative relations. But this assumption is unlikely to hold in practice, because inter-document relations will in many cases exist between two claims that were written by different authors and hence be stylistically distinct, whereas intra-document relations will in most cases be between two claims that were written by the same author, using similar terminology and styles.

In order to train robust multi-document argument mining models, high-quality annotated corpora will likely be required, where the annotations include both intra- and inter-document argumentative relationships.

The source documents included in these corpora should be tailored to the types of documents that analysts will need to analyse in practice, to ensure that the principles the model learns generalise adequately to unseen texts. This may include just one text type (such as prior intelligence reports), or a diversity of text types (chat transcripts, news articles, emails, forum posts, etc) as required. Such needs would need to be determined in collaboration with the stakeholders.

If intelligence reports form a text type of interest, then the accumulated corpus of reports generated at the Hunt Lab in response to fictional intelligence-style problems might provide quality, non-sensitive source material for annotation. The SWARM reports would likely not, however, be sufficient in volume, and would need to be supplemented with additional source material.

End-to-end argument mining As noted in Section 3.1, the pipeline approach to argument mining used here suffers from the propagation of error that occurs when each task in the pipeline is performed imperfectly. Additionally, the relation classification task is structured such that the model effectively receives no input but the text of the two claims between which it is to classify the argumentative relationship. It does not have access to import contextual information, such as the surrounding text, the authors of each claim, which documents the claims come from, whether the claims come from the same document, and if so, how proximate they are and what text segues from the first claim to the second. All this information would be of relevance to determining the relationship between too claims, and devoid of such context the classification task attempted here would be difficult for even expert humans to perform at a high level.

One promising family of solutions to this is to structure the task such that it can be completed in a single step by a single neural network [40, 22], to globally optimise the claims and relations using Integer Linear Programming [35], or to jointly train a collection of models so that their interactions are optimised and error propagation is kept to a minimum [7]. Such approaches are known as *end-to-end* argument mining. End-to-end argument mining methods have been shown to achieve significant improvements over pipeline approaches [24].

In particular, we think that articulating argument mining as a single-step task is the approach that will prove most fruitful long term. In the case of single-document argument mining, one way to articulate the task as a single step is to approach it as a sequence-tagging problem, where the tags include the indices of related components. This can be achieved by using attention modules in a neural network to determine which other segments are related. This type of network is called a *pointer network* [40]. This approach was pioneered by Potash in his PhD thesis [25], and a resulting paper [26], and has been explored further in [22].

Additional work is required to explore how multi-document argument mining can be structured as a single-step task, or a task that a single multi-task trained neural net can undertake, and what architectures are capable of learning this task.

Other improvements

- Due to limitations of time and funding, in this work we relied almost solely on previous research to inform architecture design and hyperparameter values. Future work would explore multiple model architectures and perform hyperparameter optimisation to produce more reliable models.

- Likewise due to time constraints, we did not implement simple baseline models to provide a benchmark for model performance. In this case of segmentation, such benchmarks might include a simple rule-based algortihm such as sentence segmentation. In the case of classification, majority classification would be an obvious choice. More sophisticated non-neural baselines might also be considered.

- As discussed in Section 3.5, the current method of deployment is limited with respect to the size of the corpus it is capable of analysing, on the order of 10^1 documents. This is likely too small to be of significant use in practice. Future work would explore the steps required to provide the ability to to analyse corpora consisting of 10^2 or 10^3 documents in reasonable time. Exactly what direction this work takes will depend on whether a pipeline or end-to-end approach is adopted, as the way in which each method struggles with large datasets is distinct.

- In the case of a pipeline approach, one of the key decisions to be made is how to determine which pairs of claims to pass through the classification model to assess for the existence of an argumentative relation between them. Testing all possible pairs of relations is computationally infeasible for any meaningfully sized corpus, as there are $\binom{\#\text{ claims}}{2}$ possible relations. It is an open question what heuristics to use to narrow down the set of candidates to achieve high precision and recall.

  Further, while it is clear that it is possible to structure single-document argument mining as a single step, it is unclear how such networks handle such combinatorial explosion in the number of possible connections. Understanding this may be key to successfully achieving scalable end-to-end argument mining systems.

- Currently, the system is concerned with fairly simple aspects of argumentative structure: support, attack, or equivalence relations. This ignores the structure of many real world arguments, where two or more premises may jointly imply a conclusion, but not in isolation. An example of this can be seen in [35, Fig. 1(e)]. Future work could explore how such argumentative structure could be identified by a neural network.

4 User Interface Design

4.1 Background

4.1.1 Argument visualisation

Arguments are usually communicated using dialogue or prose, but such formats are arguably not well suited to the task of conveying the precise network of claims, and the rules of inference between such claims, that constitute an argument.

There have been various attempts to formalise ways of visualising the structure of arguments, as a means to convey networks of claims and inferences more explicitly [29]. Early attempts include Wigmore charts [42], designed to visualise evidence in legal settings, and Toulmin's model of argument [36], which is more general (and not exclusively visual).

More recently, the phrase *argument map* has come to mean a graph structure, usually a tree, where each vertex corresponds to a claim, and each edge corresponds to an argumentative relationship between two claims, such as support or attack. (For clarity, we say claim a attacks claim b if a constitutes a reason to think that b is untrue.) An prototypical example of an argument map can be seen in Figure 6. Under different conventions, the propositions in an argument map can be typed (e.g. *evidence*, *assumption*, *premise*, *conclusion*) or untyped.

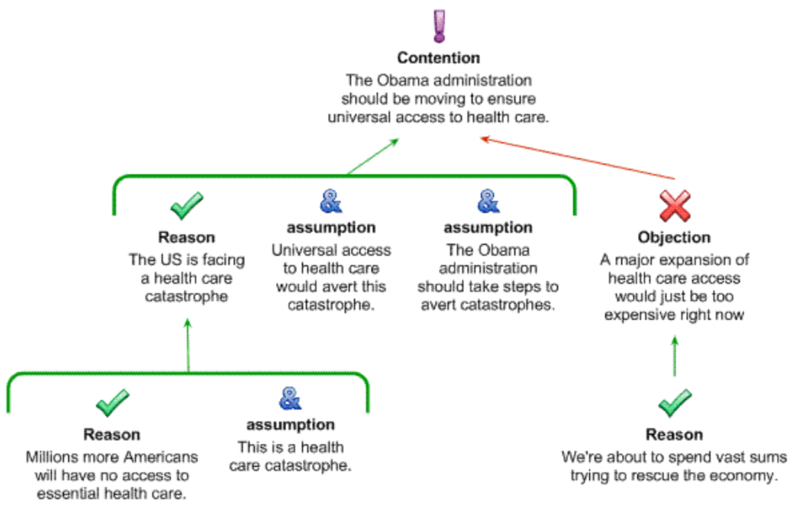

Contention
The Obama administration should be moving to ensure universal access to health care.
Reason
The US is facing a health care catastrophe
assumption
Universal access to health care would avert this catastrophe.
assumption
The Obama administration should take steps to avert catastrophes.
Objection
A major expansion of health care access would just be too expensive right now
Reason
Millions more Americans will have no access to essential health care.
assumption
This is a health care catastrophe.
Reason
We're about to spend vast sums trying to rescue the economy.

Figure 6: An example of an argument map created using the Rationale software tool, taken from [38].

A number of software tools have been produced for the purposes of visualising such tree structures. These include Araucaria [28], Rationale [39] and Kialo.

Argument maps can be useful tools to clarify the structure of arguments. But in practice, many people find them tedious to create and interpret, and this limits the extent to which they have been adopted. Additionally large argument maps can be cumbersome to navigate on average-sized screens, which further limits their accessibility.

The assumption that argumentation maps form a strict tree, which is made almost uniformly in modern argument mapping tools, is also open to question. As seen in the figure above, arguments can be often better described as so-called hi-trees [20], to capture the situation where a collection of premises jointly (but not individually) imply a conclusion. In addition, real-world argumentative prose or dialogue does often not clearly demarcate distinct claims, which can lead to loops, and both converging and diverging forks in the argument map if the map tries to remain faithful to the language of the arguments, as originally presented.

The ongoing esotericness of argument mapping suggests that there is a need for further exploration of user interfaces that make it easy and valuable to engage with such abstract structures.

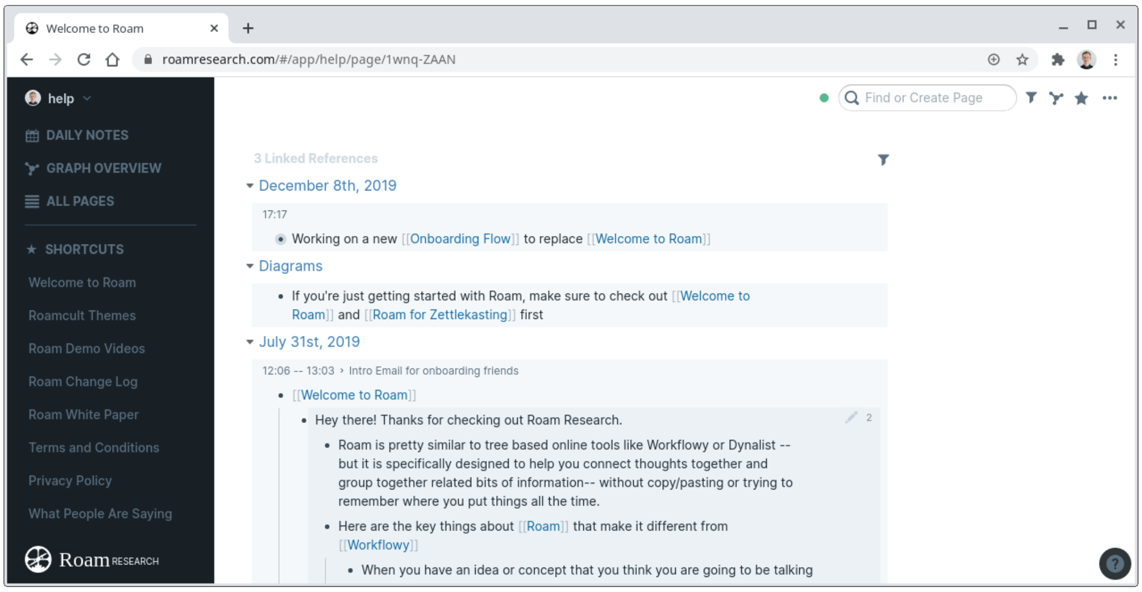
4.1.2 Networked document repositories

The rise of personal computing and the internet has led to various attempts to implement the essence of a *memex*, a hypothetical device for storing personally relevant information envisioned by Vannevar Bush in 1945 [4].

Attempts at creating this type of 'second brain' range from labyrinthine personal websites (eg. `https://www.gwern.net/`, `https://scaruffi.com/`), to personal wikis (eg. `https://tiddlywiki.com/`, `https://orgmode.org/`), to digital notebooks (eg. `https://evernote.com/`).

Over the past 18 months, there has been a resurgence of interest and innovation in these types of projects. Three new projects in particular—Roam Research, Obsidian, and the notes of Andy Matuschak—inspired the user interface ideas explored in this project.

Roam Research is a web-based note-taking application developed by Conor White-Sullivan that has developed an active and engaged user base who (jokingly) refer to themselves as the 'roamcult'. One of its key features is that of bidirectional links or *backlinks*, which refers to a section at the bottom of every note that lists all the other places in the repository where that note is referenced or linked. An example of a Roam backlinks section can be seen in Figure 7. In a diligently maintained database, this feature can be surprisingly useful in that it reminds you of connections between related concepts that may not otherwise have been front-of-mind, and hence helps you to advance your thinking more quickly.



3 Linked References

December 8th, 2019

17:17

- Working on a new [[Onboarding Flow]] to replace [[Welcome to Roam]]

Diagrams

- If you're just getting started with Roam, make sure to check out [[Welcome to Roam]] and [[Roam for Zettlekasting]] first

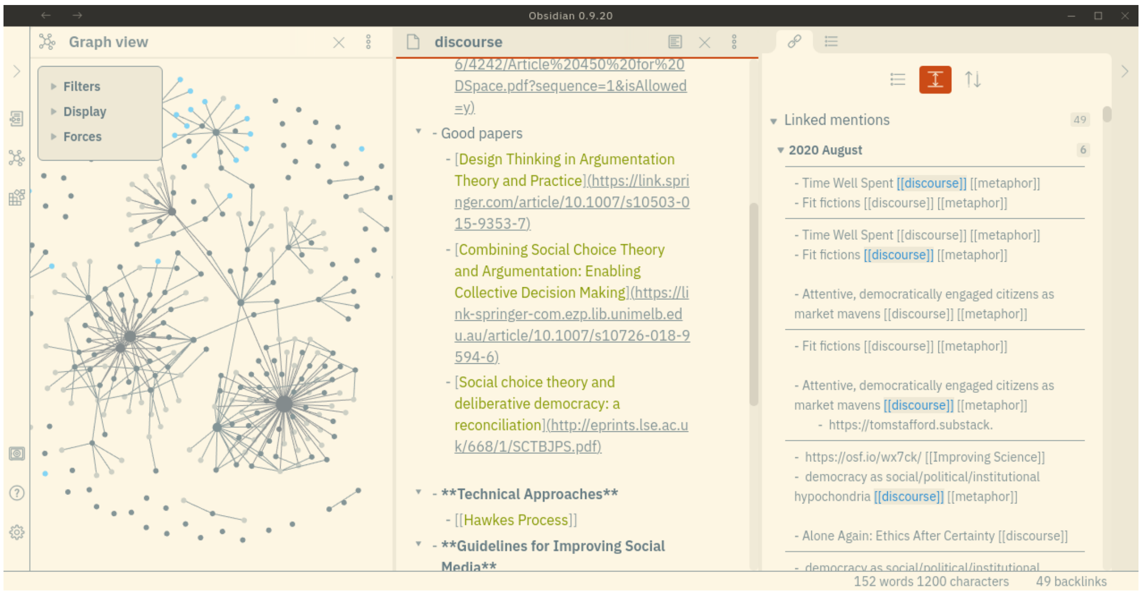
July 31st, 2019

12:06 -- 13:03 › Intro Email for onboarding friends

- [[Welcome to Roam]]
  - Hey there! Thanks for checking out Roam Research.
    - Roam is pretty similar to tree based online tools like Workflowy or Dynalist -- but it is specifically designed to help you connect thoughts together and group together related bits of information-- without copy/pasting or trying to remember where you put things all the time.
    - Here are the key things about [[Roam]] that make it different from [[Workflowy]]
      - When you have an idea or concept that you think you are going to be talking

Figure 7: An example of the backlinks section on a note in a Roam Research repository.

A similar local application is Obsidian, which stores notes as markdown files rather than in a proprietary online database. A screenshot of Obsidian can be seen in Figure 8. It also implements backlinks, as well as a graph view for visualising the structure of the repository, where notes correspond to vertices and links between notes correspond to edges.



6/4242/Article%20450%20for%20DSpace.pdf?sequence=1&isAllowed=y)
- Good papers
  - [Design Thinking in Argumentation Theory and Practice](https://link.springer.com/article/10.1007/s10503-015-9353-7)
  - [Combining Social Choice Theory and Argumentation: Enabling Collective Decision Making](https://link-springer-com.ezp.lib.unimelb.edu.au/article/10.1007/s10726-018-9594-6)
  - [Social choice theory and deliberative democracy: a reconciliation](http://eprints.lse.ac.uk/668/1/SCTBJPS.pdf)
- **Technical Approaches**
  - [[Hawkes Process]]
- **Guidelines for Improving Social Media**

Linked mentions 49

2020 August 6

- Time Well Spent [[discourse]] [[metaphor]]
- Fit fictions [[discourse]] [[metaphor]]

- Time Well Spent [[discourse]] [[metaphor]]
- Fit fictions [[discourse]] [[metaphor]]

- Attentive, democratically engaged citizens as market mavens [[discourse]] [[metaphor]]

- Fit fictions [[discourse]] [[metaphor]]

- Attentive, democratically engaged citizens as market mavens [[discourse]] [[metaphor]]
  - https://tomstafford.substack.

- https://osf.io/wx7ck/ [[Improving Science]]
- democracy as social/political/institutional hypochondria [[discourse]] [[metaphor]]

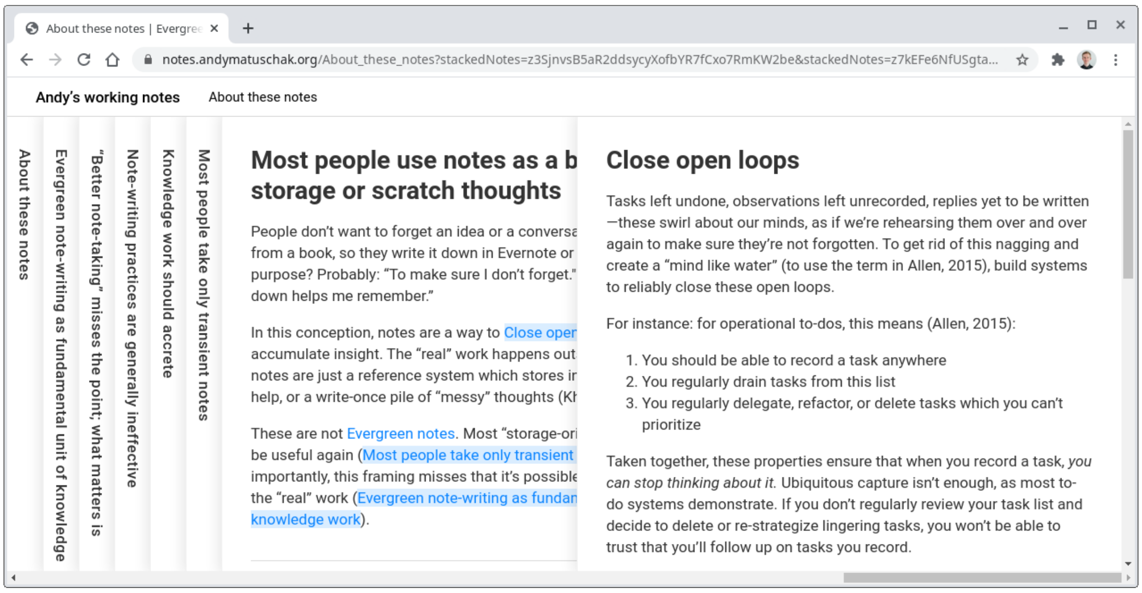
- Alone Again: Ethics After Certainty [[discourse]]

- democracy as social/political/institutional

152 words 1200 characters 49 backlinks

Figure 8: A screenshot of the Obsidian note-taking tool. The layout is customisable, but the screen shown here includes, from left to right, a visualisation of the graph structure of the repository, an editing pane for a note, and the relevant backlinks for that note.

The third influence is not a public tool, but an interface built by user interface designer Andy Matuschak for presenting his personal repository of notes (`https://notes.andymatuschak.org/`). This project also implements backlinks, and additionally makes use of a horizontal scrolling interface that facilitates exploration through the repository, making good use of the landscape orientation of most screens. See Figure 9 for a screenshot.


About these notes | Evergre
notes.andymatuschak.org/About_these_notes?stackedNotes=z3SjnvsB5aR2ddsycyXofbYR7fCxo7RmKW2be&stackedNotes=z7kEFe6NfUSgta...

Andy's working notes   About these notes

About these notes

Evergreen note-writing as fundamental unit of knowledge

"Better note-taking" misses the point; what matters is

Note-writing practices are generally ineffective

Knowledge work should accrete

Most people take only transient notes

Most people use notes as a b storage or scratch thoughts

People don't want to forget an idea or a conversa from a book, so they write it down in Evernote or purpose? Probably: "To make sure I don't forget." down helps me remember."

In this conception, notes are a way to Close oper accumulate insight. The "real" work happens out notes are just a reference system which stores ir help, or a write-once pile of "messy" thoughts (Kł

These are not Evergreen notes. Most "storage-ori be useful again (Most people take only transient importantly, this framing misses that it's possible the "real" work (Evergreen note-writing as fundan knowledge work).

Close open loops

Tasks left undone, observations left unrecorded, replies yet to be written—these swirl about our minds, as if we're rehearsing them over and over again to make sure they're not forgotten. To get rid of this nagging and create a "mind like water" (to use the term in Allen, 2015), build systems to reliably close these open loops.

For instance: for operational to-dos, this means (Allen, 2015):

1. You should be able to record a task anywhere
2. You regularly drain tasks from this list
3. You regularly delegate, refactor, or delete tasks which you can't prioritize

Taken together, these properties ensure that when you record a task, *you can stop thinking about it.* Ubiquitous capture isn't enough, as most to-do systems demonstrate. If you don't regularly review your task list and decide to delete or re-strategize lingering tasks, you won't be able to trust that you'll follow up on tasks you record.

Figure 9: A screen shot of Andy Matuschak's note repository, showing the horizontal scrolling UI.

4.2 Prototype

The prototype application built during this project is public and can be found at the following link: `https://hunt-laboratory.github.io/navigator/`.

4.2.1 Concept

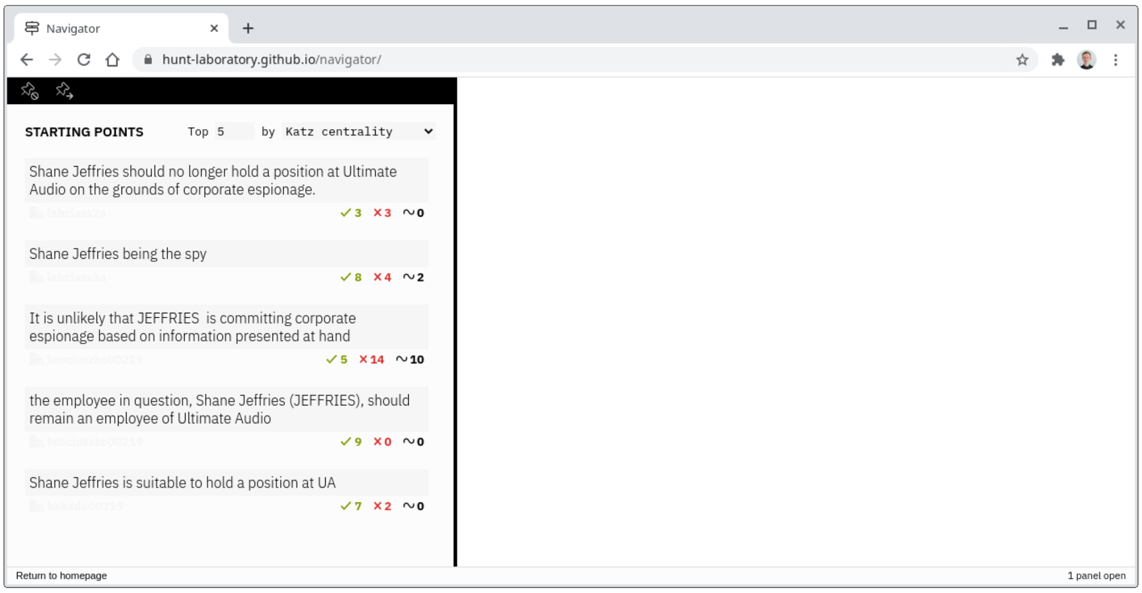
The concept for the user interface (UI) explored in this project is rooted in the correspondence between argument maps, and repositories of networked notes. Both are graph structures at heart, so innovations designed for navigating over graphs of note repositories might be fruitfully co-opted to improve interfaces for navigating large argument maps.

Specifically, this project aimed to adapt Andy Matuschak's horizontal-scrolling interface, and implement a generalised version of backlinks, where edges do not just indicate generic links, but can be typed, to separately indicate support, attack, or semantic similarity relations.

The interface is just that—an interface. It operates over a pre-existing multi-document argument map, and requires no machine learning during the exploration phase. This makes it relatively fast, as the computationally intensive argument mining can be performed in a one-off processing step at the beginning of an analysis session.

4.2.2 Features

The app opens with a single pane listing a subset of claims from the complete multi-document argument map to serve as 'starting points' (Fig. 10). By default, these are the top 5 claims as determined by the Katz centrality of those claims within the graph. The number of claims, and the metric used to sort them, are customisable. Other options for the sorting metric are the number of supporting claims, number of attacking claims, number of semantically similar claims, and the total number of connected claims.



STARTING POINTS Top 5 by Katz centrality

Shane Jeffries should no longer hold a position at Ultimate Audio on the grounds of corporate espionage.
labclass7a ✓3 ✗3 ~0

Shane Jeffries being the spy
labclass3a ✓8 ✗4 ~2

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand
kosciuszko00219 ✓5 ✗14 ~10

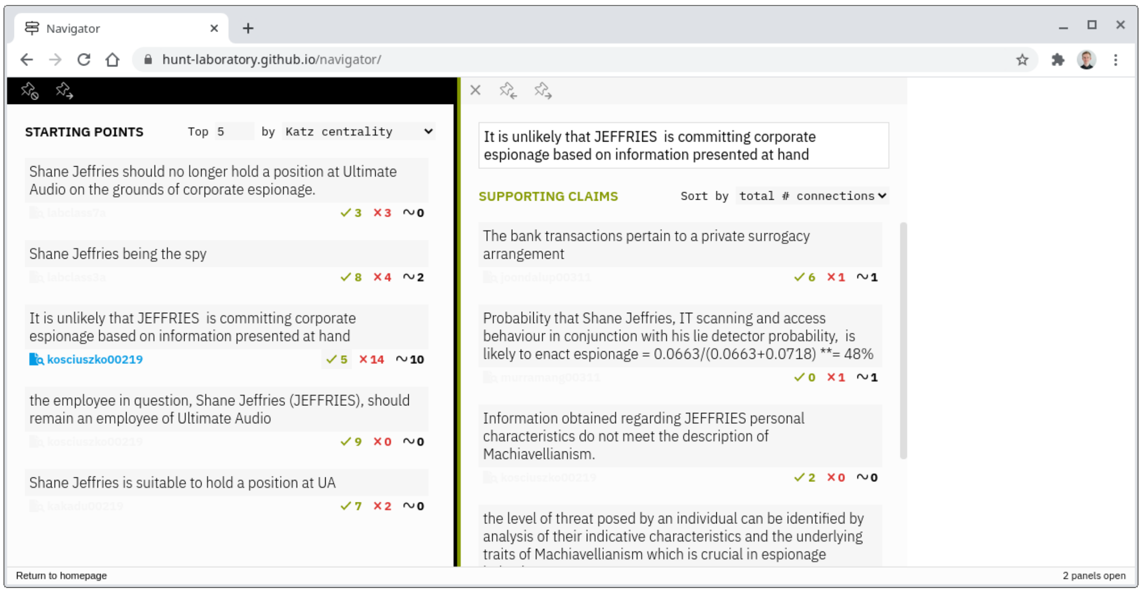
the employee in question, Shane Jeffries (JEFFRIES), should remain an employee of Ultimate Audio
kosciuszko00219 ✓9 ✗0 ~0

Shane Jeffries is suitable to hold a position at UA
kakadu00219 ✓7 ✗2 ~0

Figure 10: The opening screen of the user interface, showing a set of chosen 'starting point' claims.

Below each claim, three buttons indicate the number of claims that support, attack, and are semantically similar to that claim. These are represented by a green check mark, a red cross, and a black tilde, respectively. Clicking on any of these buttons will open a new pane, an example of which is shown in Figure 11. In the new pane, the origin claim (the claim from which you opened it) is displayed at the top. Below, all the other claims that relate to that claim in the relevant way are listed. For example, clicking on the green check mark beneath a claim will open a list of all the other claims that directly support that claim.

Using the dropdown box at the top, any list of claims can be re-ordered by any of the same five metrics used in the 'starting points' pane.


Navigator

hunt-laboratory.github.io/navigator/

STARTING POINTS Top 5 by Katz centrality

Shane Jeffries should no longer hold a position at Ultimate Audio on the grounds of corporate espionage.
labclass7a ✓3 ✗3 ~0

Shane Jeffries being the spy
labclass3a ✓8 ✗4 ~2

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand
kosciuszko00219 ✓5 ✗14 ~10

the employee in question, Shane Jeffries (JEFFRIES), should remain an employee of Ultimate Audio
kosciuszko00219 ✓9 ✗0 ~0

Shane Jeffries is suitable to hold a position at UA
kakadu00219 ✓7 ✗2 ~0

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand

SUPPORTING CLAIMS Sort by total # connections

The bank transactions pertain to a private surrogacy arrangement
joondalup00311 ✓6 ✗1 ~1

Probability that Shane Jeffries, IT scanning and access behaviour in conjunction with his lie detector probability, is likely to enact espionage = 0.0663/(0.0663+0.0718) **= 48%
murramang00311 ✓0 ✗1 ~1

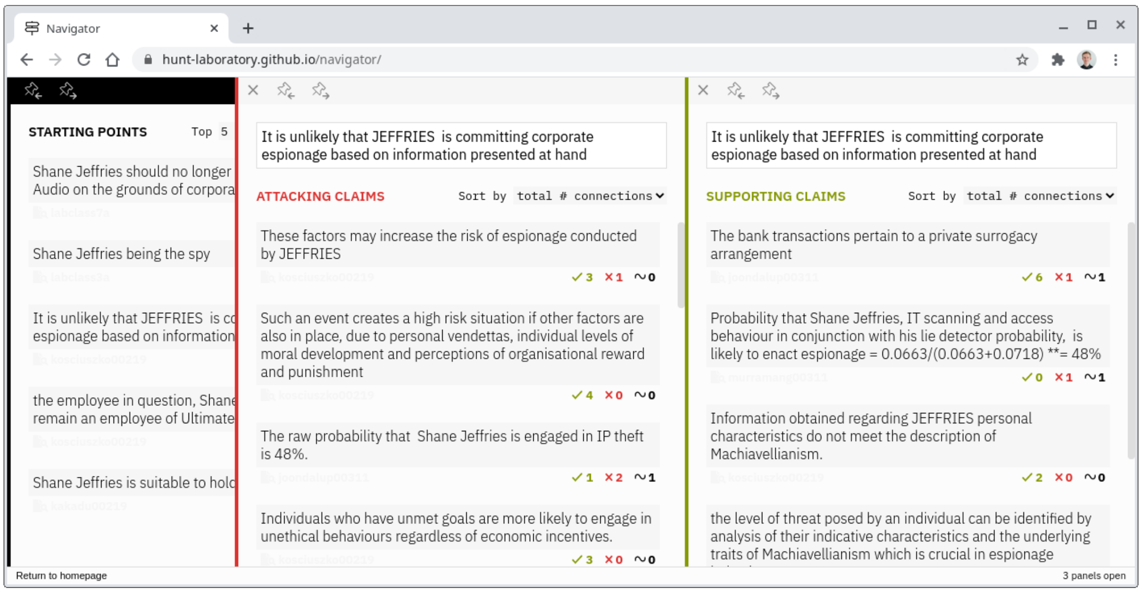
Information obtained regarding JEFFRIES personal characteristics do not meet the description of Machiavellianism.
kosciuszko00219 ✓2 ✗0 ~0

the level of threat posed by an individual can be identified by analysis of their indicative characteristics and the underlying traits of Machiavellianism which is crucial in espionage

Return to homepage 2 panels open

Figure 11: The right pane in this screenshot is an example of the standard pane type that lists all the supporting claims for a particular claim.

This workflow—noticing a claim of interest, clicking on one of the buttons beneath it, and opening new pane with a list of related claims—can be repeated *ad infinitum*. New panes will always open directly to the right of the pane from which they were opened. If a pane is no longer of interest, it can be closed using the cross in the top-left. In most cases, panes open and close with a smooth accordion-type animation, to help the user maintain visual continuity.


Navigator

hunt-laboratory.github.io/navigator/

STARTING POINTS Top 5

Shane Jeffries should no longer Audio on the grounds of corpora

Shane Jeffries being the spy

It is unlikely that JEFFRIES is co espionage based on information

the employee in question, Shane remain an employee of Ultimate

Shane Jeffries is suitable to hold

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand

ATTACKING CLAIMS Sort by total # connections

These factors may increase the risk of espionage conducted by JEFFRIES
kosciuszko00219 ✓3 ✗1 ~0

Such an event creates a high risk situation if other factors are also in place, due to personal vendettas, individual levels of moral development and perceptions of organisational reward and punishment
kosciuszko00219 ✓4 ✗0 ~0

The raw probability that Shane Jeffries is engaged in IP theft is 48%.
joondalup00311 ✓1 ✗2 ~1

Individuals who have unmet goals are more likely to engage in unethical behaviours regardless of economic incentives.
kosciuszko00219 ✓3 ✗0 ~0

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand

SUPPORTING CLAIMS Sort by total # connections

The bank transactions pertain to a private surrogacy arrangement
joondalup00311 ✓6 ✗1 ~1

Probability that Shane Jeffries, IT scanning and access behaviour in conjunction with his lie detector probability, is likely to enact espionage = 0.0663/(0.0663+0.0718) **= 48%
murramang00311 ✓0 ✗1 ~1

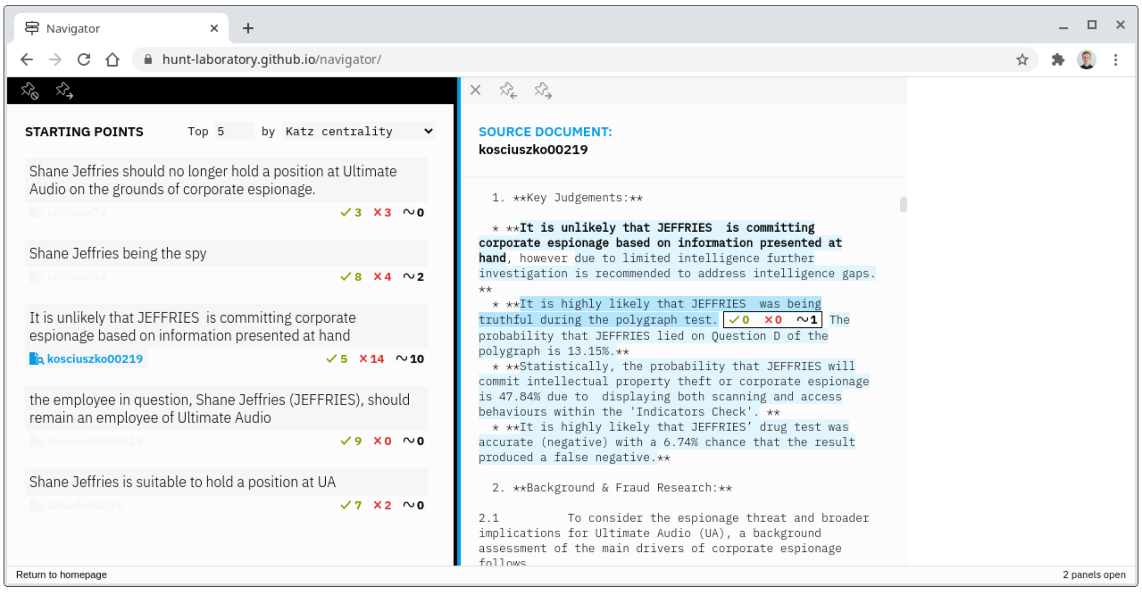
Information obtained regarding JEFFRIES personal characteristics do not meet the description of Machiavellianism.
kosciuszko00219 ✓2 ✗0 ~0

the level of threat posed by an individual can be identified by analysis of their indicative characteristics and the underlying traits of Machiavellianism which is crucial in espionage

Return to homepage 3 panels open

Figure 12: A second pane opened, this time listing attacking claims. As more panes are opened, they will begin to overlap previous panes, but the user can scroll left and right using a two-fingered swipe on a touchpad, or by holding down `Shift` and using the scroll wheel on their mouse.

Often, it will be useful to view a particular claim in the context of the document from which it originates. When hovering over any given claim, the name of the origin document will appear in blue. Clicking on the name will open a pane with the text of the original document. The document will autoscroll so that the claim of interest is visible, and the claim will appear in bold. All claims in the document will be highlighted in pale blue. Clicking on any of the claims will append it with the standard set of three buttons, or generalised backlinks, that allow the user to open lists of claims that are related to that claim. This can be seen in Figure 13.


Navigator
hunt-laboratory.github.io/navigator/

STARTING POINTS Top 5 by Katz centrality

Shane Jeffries should no longer hold a position at Ultimate Audio on the grounds of corporate espionage.
labclass7a ✓3 ✗3 ∼0

Shane Jeffries being the spy
labclass3a ✓8 ✗4 ∼2

It is unlikely that JEFFRIES is committing corporate espionage based on information presented at hand
kosciuszko00219 ✓5 ✗14 ∼10

the employee in question, Shane Jeffries (JEFFRIES), should remain an employee of Ultimate Audio
kosciuszko00219 ✓9 ✗0 ∼0

Shane Jeffries is suitable to hold a position at UA
kakadu00219 ✓7 ✗2 ∼0

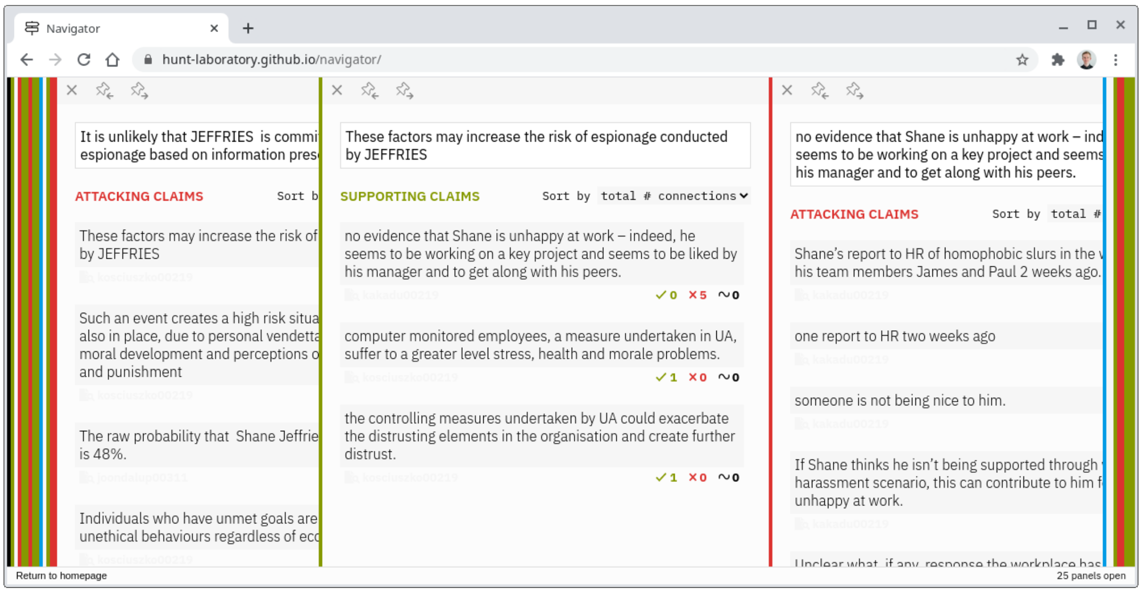
SOURCE DOCUMENT:
kosciuszko00219

```
  1. **Key Judgements:**

  * **It is unlikely that JEFFRIES  is committing
corporate espionage based on information presented at
hand, however due to limited intelligence further
investigation is recommended to address intelligence gaps.
**
  * **It is highly likely that JEFFRIES  was being
truthful during the polygraph test. ✓0 ✗0 ∼1 The
probability that JEFFRIES lied on Question D of the
polygraph is 13.15%.**
  * **Statistically, the probability that JEFFRIES will
commit intellectual property theft or corporate espionage
is 47.84% due to  displaying both scanning and access
behaviours within the 'Indicators Check'. **
  * **It is highly likely that JEFFRIES' drug test was
accurate (negative) with a 6.74% chance that the result
produced a false negative.**

  2. **Background & Fraud Research:**

2.1         To consider the espionage threat and broader
implications for Ultimate Audio (UA), a background
assessment of the main drivers of corporate espionage
follows
```

Return to homepage
2 panels open

Figure 13: An example of a pane that presents a claim in the context of the document from which it has been extracted. All other claims in the document are highlighted in blue. Click on any one will open the standard three buttons that indicate argumentative relations.

As the number of open panes grows large, it can be difficult to keep track of context and your location within the argument map. To some extent, doing so is not a priority: this is designed to be an exploratory tool that can surface ideas and evidence that require follow-up analysis. But one small feature is designed to help the user keep track of their location: open panes will 'stack up' at the left or right edge of the screen, and their coloured border will be visible even when their text is obscured from view. This provides a subtle indication of location, and the sequence of stacked border colours loosely tracks the path through which you arrived at the current pane.


Navigator
hunt-laboratory.github.io/navigator/

It is unlikely that JEFFRIES is commi
espionage based on information pres

ATTACKING CLAIMS Sort b

These factors may increase the risk of
by JEFFRIES
kosciuszko00219

Such an event creates a high risk situa
also in place, due to personal vendetta
moral development and perceptions o
and punishment
kosciuszko00219

The raw probability that Shane Jeffrie
is 48%.
joondalup00311

Individuals who have unmet goals are
unethical behaviours regardless of ec
kosciuszko00219

These factors may increase the risk of espionage conducted by JEFFRIES

SUPPORTING CLAIMS Sort by total # connections

no evidence that Shane is unhappy at work – indeed, he seems to be working on a key project and seems to be liked by his manager and to get along with his peers.
kakadu00219 ✓0 ✗5 ∼0

computer monitored employees, a measure undertaken in UA, suffer to a greater level stress, health and morale problems.
kosciuszko00219 ✓1 ✗0 ∼0

the controlling measures undertaken by UA could exacerbate the distrusting elements in the organisation and create further distrust.
kosciuszko00219 ✓1 ✗0 ∼0

no evidence that Shane is unhappy at work – ind
seems to be working on a key project and seems
his manager and to get along with his peers.

ATTACKING CLAIMS Sort by total #

Shane's report to HR of homophobic slurs in the
his team members James and Paul 2 weeks ago.
kakadu00219

one report to HR two weeks ago
kakadu00219

someone is not being nice to him.
kakadu00219

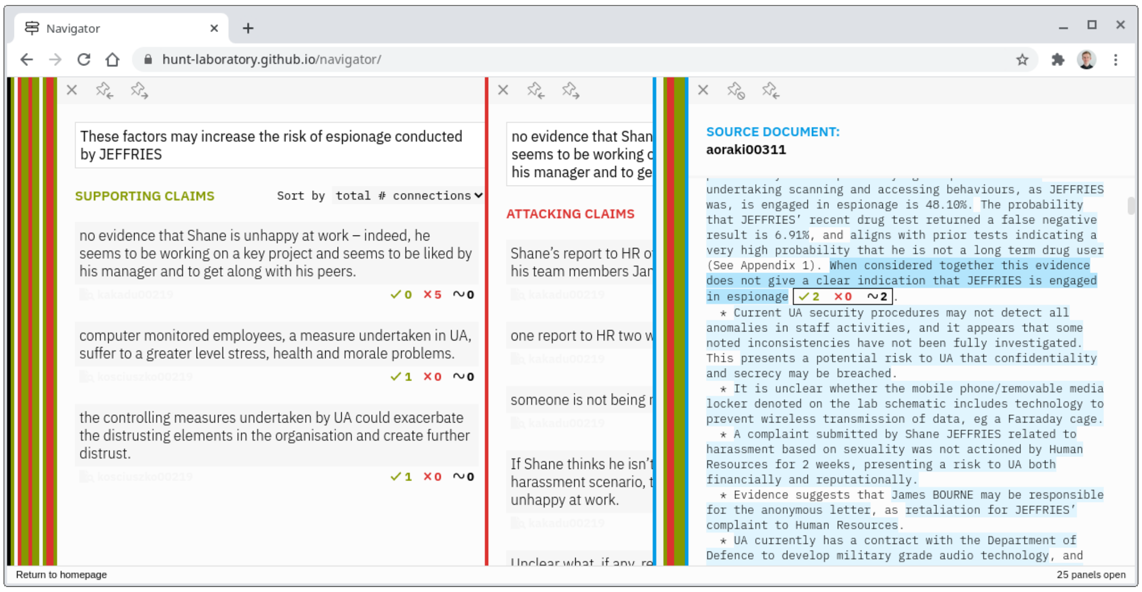
If Shane thinks he isn't being supported through
harassment scenario, this can contribute to him f
unhappy at work.
kakadu00219

Unclear what, if any, response the workplace has

Return to homepage
25 panels open

Figure 14: When there are many panes open, they will stack up at the left and right of the screen.

A final usability feature is the ability to pin panes to the left or right of the screen. This will take them out of the usual flow of panes, so that they are always visible as a reference point. An example of this can be seen in Figure 15, where a source document has been pinned to the right.



Navigator

hunt-laboratory.github.io/navigator/

These factors may increase the risk of espionage conducted by JEFFRIES

SUPPORTING CLAIMS   Sort by total # connections

no evidence that Shane is unhappy at work – indeed, he seems to be working on a key project and seems to be liked by his manager and to get along with his peers.
kakadu00219   ✓0 ✗5 ∼0

computer monitored employees, a measure undertaken in UA, suffer to a greater level stress, health and morale problems.
kosciuszko00219   ✓1 ✗0 ∼0

the controlling measures undertaken by UA could exacerbate the distrusting elements in the organisation and create further distrust.
kosciuszko00219   ✓1 ✗0 ∼0

no evidence that Shan
seems to be working o
his manager and to ge

ATTACKING CLAIMS

Shane's report to HR o
his team members Jar
kakadu00219

one report to HR two w
kakadu00219

someone is not being r
kakadu00219

If Shane thinks he isn't
harassment scenario, t
unhappy at work.
kakadu00219

Unclear what, if any, re

SOURCE DOCUMENT:
aoraki00311

```
undertaking scanning and accessing behaviours, as JEFFRIES
was, is engaged in espionage is 48.10%. The probability
that JEFFRIES' recent drug test returned a false negative
result is 6.91%, and aligns with prior tests indicating a
very high probability that he is not a long term drug user
(See Appendix 1). When considered together this evidence
does not give a clear indication that JEFFRIES is engaged
in espionage ✓2 ✗0 ∼2.
  * Current UA security procedures may not detect all
anomalies in staff activities, and it appears that some
noted inconsistencies have not been fully investigated.
This presents a potential risk to UA that confidentiality
and secrecy may be breached.
  * It is unclear whether the mobile phone/removable media
locker denoted on the lab schematic includes technology to
prevent wireless transmission of data, eg a Farraday cage.
  * A complaint submitted by Shane JEFFRIES related to
harassment based on sexuality was not actioned by Human
Resources for 2 weeks, presenting a risk to UA both
financially and reputationally.
  * Evidence suggests that James BOURNE may be responsible
for the anonymous letter, as retaliation for JEFFRIES'
complaint to Human Resources.
  * UA currently has a contract with the Department of
Defence to develop military grade audio technology, and
```

Return to homepage

25 panels open

Figure 15: A blue source document pane, pinned to the right of the screen.

By default, the black 'starting points' pane is pinned to the left of the screen, but can be pinned and unpinned freely, as can any other pane.

4.2.3 Technical details

The UI is implemented as a client-side, browser-based web application, written in JavaScript using the neverland framework (`https://github.com/WebReflection/neverland`). The neverland framework has very similar functionality to other popular web development frameworks such as React, but is quicker, much smaller, and does not need to be compiled as it only requires vanilla JavaScript.

For demonstration purposes, the application is hosted in a public GitHub repository and made accessible as a GitHub Pages site.

4.3 Future work

The interface is quite usable as is, but could be made more so in a number of ways. Below, we list directions for future work.

- Provide the ability to save the state of the tools, so that it can be closed and re-opened to the same pane configuration, without re-analysing the texts.
- Aggregate semantically similar claims within any list, so that that multiple semantically similar claims do not misleadingly appear as distinct.
- Provide mechanisms for searching and filtering claims in useful ways, such as by user-added tag or source document.
- Provide buttons to access argumentative relations in both directions. Currently, the interface only gives the ability to view *incoming* relations. For example, all the claims that support a given node, but not the claims that it, in turn, supports. It would be useful to provide the other direction as well so that users can navigate both forwards and backwards through the graph.

- Figure out ways to better preserve context. At the moment, it can be difficult to keep track of context and history when there is a large number of panes open. To some extent this is an example of a fundamental problem with ordering ideas - networks of concepts tend to be fractal, so forcing them into a linear sequence will inevitably mean that some related concepts end up far apart. But there are directions to explore that may help. For example, simply allowing the user to save or star claims of particular interest would allow them to be relocated more easily. A deeper rebuild could explore a treemap layout of panes on large screens, which would better be able to reflect the structure of the underlying argument map.

- As noted earlier, real world arguments often take on a hi-tree structure [20]. A more general interface would have the ability to reflect such structure, indicating how multiple claims jointly support or attack another claim, where none of those claims does so individually.

- The interface is already relatively quick, but there is some unnecessary processing (re-searching for relevant claims that have already been found) that could be obviated with some work to optimise the back end of the interface. This would allow the interface to remain responsive when exploring argument maps that are orders of magnitude larger than those considered in this project.

- Before rolling out any new software to a workforce, it is important to have confidence that the software will be user friendly and an improvement on existing systems. This is particularly true for intelligence organisations, where it has been noted that new tools for analysis frequently fail to be adopted because they are designed by outsiders with little consideration for how they fit into analysts' existing workflows. For instance, see the following excerpt from Hoffman et al. [9].

  > In effect, the tools are instances of the trap of designer-centred design. This trap is hard to avoid – smart and well-intentioned people develop their own theory of the work to be supported, come to believe they can 'fill the shoes' of the target users, and then build a tool that is, in effect, an hypothesis of how the work will change once the tool is introduced. Generally, the hypothesis is that work will be improved, but there are almost always certain kinds of unanticipated consequences. Users have to fight the technology and are prevented from being fully engaged in the problem. As a result, they often end-up creating workarounds or kluges such that they can progress their work in spite of the technology (Koopman and Hoffman 2003, Moon and Hoffman 2005, Laplante et al. 2007).
  >
  > When a support tool is presented for consideration, one must ask these kinds of questions: Was any form of cognitive task analysis undertaken to understand the user's current work methods, challenges and bottlenecks? What model of work does the tool support? How involved were analysts in the tool development process? Is there convincing empirical evidence that the tool is useful and understandable? Once a new tool is in place, is the changed work observable such that performance and tool effectiveness and impact can be studied?

  Sincere development of Navigator would involve usability testing, analyst feedback, and well-designed experiments to test whether the software truly improves analysts' ability to extract value from large document corpora.

5 Acknowledgements

This research was improved with feedback from other members of the Hunt Lab at the University of Melbourne. In alphabetical order: Ashley Barnett, Ariel Kruger, Tamar Primoratz, Richard de Rozario, Morgan Saletta and Tim van Gelder.

It also benefitted from conversations the author had with Jacky Visser and Rory Duthie from the Centre for Argument Technology at the University of Dundee.

The initiative was funded by the Department of Defence and the Office of National Intelligence under the AI for Decision Making Program, delivered in partnership with the Defence Science Institute in Victoria.

6 References

[1] Yamen Ajjour et al. "Unit segmentation of argumentative texts". In: *Proceedings of the 4th Workshop on Argument Mining*. 2017, pp. 118–128.

[2] Roy Bar-Haim et al. "Quantitative Argument Summarization and Beyond: Cross-Domain Key Point Analysis". In: *arXiv preprint arXiv:2010.05369* (2020).

[3] Roy Bar-Haim et al. "Stance classification of context-dependent claims". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 2017, pp. 251–261.

[4] Vannevar Bush et al. "As we may think". In: *The atlantic monthly* 176.1 (1945), pp. 101–108.

[5] Artem Chernodub et al. "Targer: Neural argument mining at your fingertips". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2019, pp. 195–200.

[6] Oana Cocarascu et al. "A dataset independent set of baselines for relation prediction in argument mining". In: *arXiv preprint arXiv:2003.04970* (2020).

[7] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. "Neural end-to-end learning for computational argumentation mining". In: *arXiv preprint arXiv:1704.06104* (2017).

[8] Tim van Gelder, Richard de Rozario, and Morgan Saletta. "Crowdsourcing Citizen Intelligence". In: *Proceedings of the Technology Surprise Forum, Safeguarding Australia 2019*. National Security College, The Australian National University, & Defence Science and Technology (DST) Group, Australian Department of Defence. (Eds.), 2019. URL: https://bit.ly/3h1o82c.

[9] Robert Hoffman et al. "Reasoning difficulty in analytical activity". In: *Theoretical Issues in Ergonomics Science* 12.3 (2011), pp. 225–240.

[10] Mathilde Janier, John Lawrence, and Chris Reed. "OVA+: An argument analysis interface". In: *Computational Models of Argument: Proceedings of COMMA*. Vol. 266. 2014. 2014, p. 463.

[11] Leo Katz. "A new status index derived from sociometric analysis". In: *Psychometrika* 18.1 (1953), pp. 39–43.

[12] Marcin Koszowy et al. "The Role of Rephrase in Argumentation: Computational and Cognitive Aspects". In: ().

[13] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[14] Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. "An argument-annotated corpus of scientific publications". In: *Proceedings of the 5th Workshop on Argument Mining*. 2018, pp. 40–46.

[15] John Lawrence and Chris Reed. "Argument mining: A survey". In: *Computational Linguistics* 45.4 (2020), pp. 765–818.

[16] John Lawrence and Chris Reed. "Combining argument mining techniques". In: *Proceedings of the 2nd Workshop on Argumentation Mining*. 2015, pp. 127–136.

[17] John Lawrence et al. "Mining arguments from 19th century philosophical texts using topic based modelling". In: (2014).

[18] Marco Lippi and Paolo Torroni. "Argumentation mining: State of the art and emerging trends". In: *ACM Transactions on Internet Technology (TOIT)* 16.2 (2016), pp. 1–25.

[19] Xuezhe Ma and Eduard Hovy. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1064–1074. DOI: 10.18653/v1/P16-1101.

[20] Kim Marriott et al. "Hi-trees and their layout". In: *IEEE Transactions on Visualization and Computer Graphics* 17.3 (2010), pp. 290–304.

[21] Marie-Francine Moens et al. “Automatic detection of arguments in legal texts”. In: *Proceedings of the 11th international conference on Artificial intelligence and law.* 2007, pp. 225–230.

[22] Gaku Morio and Katsuhide Fujita. “End-to-end argument mining for discussion threads based on parallel constrained pointer architecture”. In: *arXiv preprint arXiv:1809.00563* (2018).

[23] Bridget Rose Nolan. “Information sharing and collaboration in the United States Intelligence community: an ethnographic study of the National Counterterrorism Center”. PhD thesis. University of Pennsylvania, 2013.

[24] Isaac Persing and Vincent Ng. “End-to-end argumentation mining in student essays”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* 2016, pp. 1384–1394.

[25] Peter Potash. “Neural argumentation: Structure and persuasion”. PhD thesis. University of Massachusetts Lowell, 2017.

[26] Peter Potash, Alexey Romanov, and Anna Rumshisky. “Here’s my point: Joint pointer architecture for argument mining”. In: *arXiv preprint arXiv:1612.08994* (2016).

[27] Lance A Ramshaw and Mitchell P Marcus. “Text chunking using transformation-based learning”. In: *Natural language processing using very large corpora.* Springer, 1999, pp. 157–176.

[28] Chris Reed and Glenn Rowe. “Araucaria: Software for argument analysis, diagramming and representation”. In: *International Journal on Artificial Intelligence Tools* 13.04 (2004), pp. 961–979.

[29] Chris Reed, Douglas Walton, and Fabrizio Macagno. “Argument diagramming in logic, law and artificial intelligence”. In: *Knowledge Engineering Review* 22.1 (2007), pp. 87–110.

[30] Nils Reimers et al. “Classification and clustering of arguments with contextualized word embeddings”. In: *arXiv preprint arXiv:1906.09821* (2019).

[31] Ruty Rinott et al. “Show me your evidence-an automatic method for context dependent evidence detection”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing.* 2015, pp. 440–450.

[32] Christos Sardianos et al. “Argument extraction from news”. In: *Proceedings of the 2nd Workshop on Argumentation Mining.* 2015, pp. 56–66.

[33] Christian Stab and Iryna Gurevych. “Parsing argumentation structures in persuasive essays”. In: *Computational Linguistics* 43.3 (2017), pp. 619–659.

[34] Christian Stab, Tristan Miller, and Iryna Gurevych. “Cross-topic argument mining from heterogeneous sources using attention-based neural networks”. In: *arXiv preprint arXiv:1802.05758* (2018).

[35] Christian Matthias Edwin Stab. “Argumentative writing support by means of natural language processing”. PhD thesis. Technische Universität Darmstadt, 2017.

[36] Stephen E Toulmin. *The uses of argument.* Cambridge university press, 2003.

[37] Dietrich Trautmann et al. “Fine-Grained Argument Unit Recognition and Classification.” In: *AAAI.* 2020, pp. 9048–9056.

[38] Tim Van Gelder. “Argument mapping”. In: *Encyclopedia of the Mind* 1 (2013), pp. 51–53.

[39] Tim Van Gelder. “The rationale for Rationale™”. In: *Law, probability and risk* 6.1-4 (2007), pp. 23–42.

[40] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. “Pointer networks”. In: *Advances in neural information processing systems.* 2015, pp. 2692–2700.

[41] Simon Wells. “Argument mining: Was ist das?” In: *CMNA14* (2014).

[42] John H Wigmore. “Problem of proof”. In: *Ill. LR* 8 (1913), p. 77.