Argument Discovery and Extraction with the Argument Workbench

Adam Wyner
Computing Science
University of Aberdeen
Aberdeen, United Kingdom
azwyner@abdn.ac.uk

Wim Peters
Computer Science
University of Sheffield
Sheffield, United Kingdom
w.peters@sheffield.ac.uk

David Price
DebateGraph
United Kingdom
david@debategraph.org

Abstract

The paper discusses the architecture and development of an Argument Workbench, which is a interactive, integrated, modular tool set to extract, reconstruct, and visualise arguments. We consider a corpora with dispersed information across texts, making it essential to conceptually search for argument elements, topics, and terminology. The Argument Workbench is a processing cascade, developed in collaboration with DebateGraph. The tool supports an argument engineer to reconstruct arguments from textual sources, using information processed at one stage as input to a subsequent stage of analysis, and then building an argument graph. We harvest and preprocess comments; highlight argument indicators, speech act and epistemic terminology; model topics; and identify domain terminology. We use conceptual semantic search over the corpus to extract sentences relative to argument and domain terminology. The argument engineer uses the extracts for the construction of arguments in DebateGraph.

1 Introduction

Argumentative text is rich, multidimensional, and fine-grained, consisting of (among others): a range of (explicit and implicit) discourse relations between statements in the corpus, including indicators for conclusions and premises; speech acts and propositional attitudes; contrasting sentiment terminology; and domain terminology. Moreover, linguistic expression is various, given alternative syntactic or lexical forms for related semantic meaning. It is difficult for humans to reconstruct argument from text, let alone for a computer. This is especially the case where arguments are dispersed across unstructured textual corpora. In our view, the most productive scenario is one in which a human argument engineer is maximally assisted in her work by computational means in the form of automated text filtering and annotation. This enables the engineer to focus on text that matters and further explore the argumentation structure on the basis of the added metadata. The Argument WorkBench (AWB) captures this process of incremental refinement and extension of the argument structure, which the engineer then produces as a structured object with a visual representation.

Given the abundance of textual source data available for argumentation analysis there is a real need for automated filtering and interpretation. Current social media platforms provide an unprecedented source of user-contributed content on most any topic. Reader-contributed comments to a comment forum, e.g. for a news article, are a source of arguments for and against issues raised in the article, where an argument is a claim with justifications and exceptions. It is difficult to coherently understand the overall, integrated meaning of the comments.

To reconstruct the arguments sensibly and reusably, we build on a prototype Argument Workbench (AWB) (Wyner et al.(2012); Wyner(2015)), which is a semi-automated, interactive, integrated, modular tool set to extract, reconstruct, and visualise arguments. The workbench is a processing cascade, developed in collaboration with an industrial partner

Proceedings of the 2nd Workshop on Argumentation Mining, pages 78–83,
Denver, Colorado, June 4, 2015. ©2015 Association for Computational Linguistics

DebateGraph and used by an Argumentation Engineer, where information processed at one stage gives greater structure for the subsequent stage. In particular, we: harvest and pre-process comments; highlight argument indicators, speech act terminology, epistemic terminology; model topics; and identify domain terminology and relationships. We use conceptual semantic search over the corpus to extract sentences relative to argument and domain terminology. The argument engineer analyses the output and then inputs extracts into the DebateGraph visualisation tool. The novelty of the work presented in this paper is the addition of terminology (domain topics and key words, speech act, and epistemic) along with the workflow analysis provided by our industrial partner. For this paper, we worked with a corpus of texts bearing on the Scottish Independence vote in 2014; however, the tool is neutral with respect to domain, since the domain terminology is derived using automatic tools.

In this short paper, we briefly outline the AWB workflow, sketch tool components, provide sample query results, discuss related work in the area, and close with a brief discussion.

2 The Argument WorkBench Workflow

The main user of the Argument WorkBench (AWB) is Argumentation Engineer, an expert in argumentation modeling who uses the Workbench to select and interpret the text material. Although the AWB automates some of the subtasks involved, the ultimate modeler is the argumentation engineer. The AWB distinguishes between the selection and modeling tasks, where selection is computer-assisted and semi-automatic, whereas the modeling is performed manually in DebateGraph (see Figure 1).

The AWB encompasses a flexible methodology that provides a workflow and an associated set of modules that together form a flexible and extendable methodology for the detection of argument in text. Automated techniques provide textually grounded information about conceptual nature of the domain and the argument structure by means of the detection of argument indicators. This information, in the form of textual metadata, enable the argumentation engineer to filter out potentially interesting text for eventual manual analysis, validation and evaluation.

Figure 1 shows the overall workflow. Document collection is not taken into account. In the first stage, text analysis such as topic, term and named entity extraction provides a first thematic grouping and semantic classification of relevant domain elements. This combination of topics, named entities and terms automatically provides the first version of a domain model, which assists the engineer in the conceptual interpretation and subsequent exploration. The texts filtered in this thematic way can then be filtered further with respect to argument indicators (discourse terminology, speech acts, epistemic terminology) as well as sentiment (positive and negative terminology). At each stage, the Argumentation Engineer is able to query the corpus with respect to the metadata (which we also refer to as the conceptual annotations). This complex filtering of information from across a corpus helps the Argumentation Engineer consolidate her understanding of the argumentative role of information.

3 AWB Components

3.1 Text Analysis

To identify and extract the textual elements from the source material, we use the GATE framework (Cunningham et al.(2002)) for the production of semantic metadata in the form of annotations.

GATE is a framework for language engineering applications, which supports efficient and robust text processing including functionality for both manual and automatic annotation (Cunningham et al.(2002)); it is highly scalable and has been applied in many large text processing projects; it is an open source desktop application written in Java that provides a user interface for professional linguists and text engineers to bring together a wide variety of natural language processing tools and apply them to a set of documents. The tools are concatenated into a pipeline of natural language processing modules. The main modules we are using in our bottom-up and incremental tool development (Wyner and Peters(2011)) perform the following functionalities:

- linguistic pre-processing. Texts are segmented into tokens and sentences; words are assigned Part-of-Speech (POS).

- gazetteer lookup. A gazetteer is a list of words

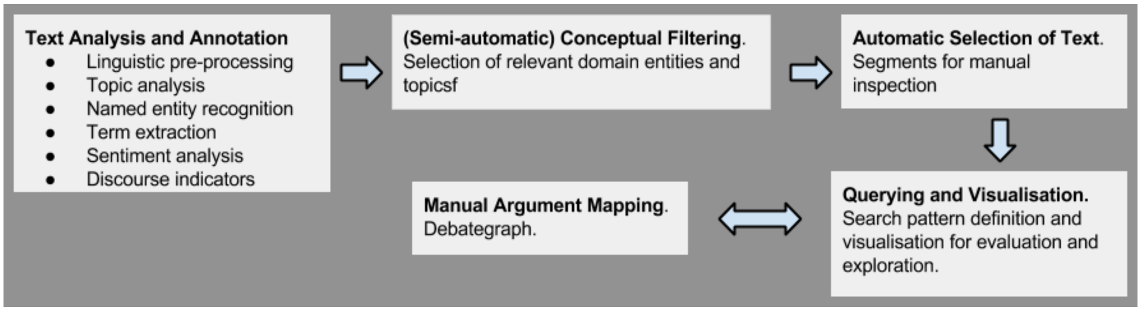

Text Analysis and Annotation
Linguistic pre-processing
Topic analysis
Named entity recognition
Term extraction
Sentiment analysis
Discourse indicators
(Semi-automatic) Conceptual Filtering.
Selection of relevant domain entities and topicsf
Automatic Selection of Text.
Segments for manual inspection
Manual Argument Mapping.
Debategraph.
Querying and Visualisation.
Search pattern definition and visualisation for evaluation and exploration.

Figure 1: Overview of the Argument WorkBench Workflow

associated with a central concept. In the lookup phase, text in the corpus is matched with terms on the lists, then assigned an annotation.

- annotation assignment through rule-based grammars, where rules take annotations and regular expressions as input and produce annotations as output.

Once a GATE pipeline has been applied, the argument engineer views the annotations *in situ* or using GATE's ANNIC (ANNotations In Context) corpus indexing and querying tool (see section 4), which enables semantic search for annotation patterns across a distributed corpus.

3.2 Term and Topic Extraction

In the current version of the AWB, we used two automatic approches to developing terminology, allowing the tool to be domain independent and rapidly developed. We used the *TermRaider* tool in GATE to identify relevant terminology (Maynard et al.(2008)). TermRaider automatically provides domain-specific noun phrase term candidates from a text corpus together with a statistically derived termhood score. Possible terms are filtered by means of a multi-word-unit grammar that defines the possible sequences of part of speech tags constituting noun phrases. It computes term frequency/inverted document frequency (TF/IDF), which takes into account term frequency and the number of documents in the collection, yielding a score that indicates the salience of each term candidate for each document in the corpus. All term candidates with a TF/IDF score higher than an manually determined threshold are then selected and presented as candidate relevant terms, annotated as such in the corpus. In addition to TermRaider, we have used a tool to model topics, identifying clusters of terminology that are taken to statistically "cohere" around a topic; for this, we have used a tool based on *Latent Dirichlet Allocation* (Blei et al.(2008)). Each word in a topic is used to annotate every sentence in the corpus that contains that word. Thus, with term and topic annotation, the Argumentation Engineer is able to query the corpus for relevant, candidate passages.

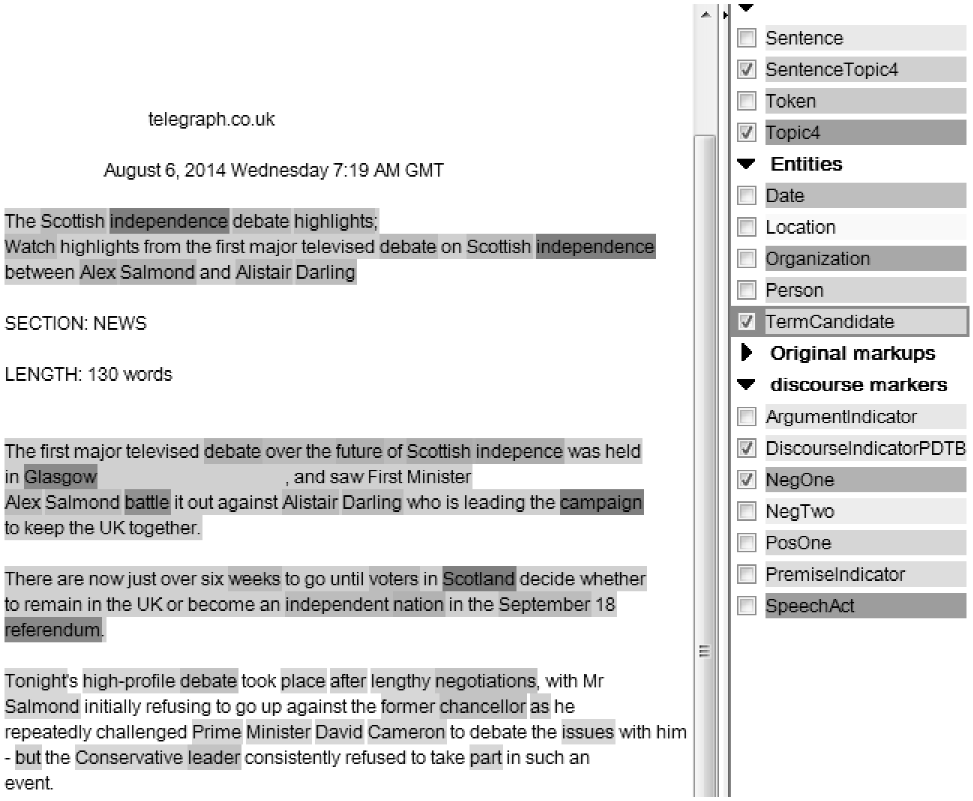
3.3 DebateGraph

DebateGraph is a free, cloud-based platform that enables communities of any size to build and share dynamic interactive visualizations of all the ideas, arguments, evidence, options and actions that anyone in the community believes relevant to the issues under consideration, and to ensure that all perspectives are represented transparently, fairly, and fully in a meaningful, structured and iterative dialogue. It supports formal argumentation as well as structured dialogue, and has been used by, amongst others, CNN, The Independent newspaper, the White House Office of Science and Technology Policy, the European Commission, and the UK's Prime Minister's Office as well as the Foreign and Commonwealth Office.

4 Viewing and Querying

The AWB enriches the manual, close reading oriented method of argument map creation in DebateGraph with automated analysis, which filters relevant text segments with respect to a certain topic of interest, and provides initial argument structure information to the text by means of annotations.

Once the corpus is annotated, we can view the annotations in the documents themselves. In Figure 2, we have a text that has been highlighted with



telegraph.co.uk

August 6, 2014 Wednesday 7:19 AM GMT

The Scottish independence debate highlights;
Watch highlights from the first major televised debate on Scottish independence between Alex Salmond and Alistair Darling

SECTION: NEWS

LENGTH: 130 words

The first major televised debate over the future of Scottish indepence was held in Glasgow , and saw First Minister Alex Salmond battle it out against Alistair Darling who is leading the campaign to keep the UK together.

There are now just over six weeks to go until voters in Scotland decide whether to remain in the UK or become an independent nation in the September 18 referendum.

Tonight's high-profile debate took place after lengthy negotiations, with Mr Salmond initially refusing to go up against the former chancellor as he repeatedly challenged Prime Minister David Cameron to debate the issues with him - but the Conservative leader consistently refused to take part in such an event.

□ Sentence
☑ SentenceTopic4
□ Token
☑ Topic4

Entities

□ Date
□ Location
□ Organization
□ Person
☑ TermCandidate

Original markups

discourse markers

□ ArgumentIndicator
☑ DiscourseIndicatorPDTB
☑ NegOne
□ NegTwo
□ PosOne
□ PremiseIndicator
□ SpeechAct

Figure 2: Highlighting Annotations in the Text

a selection of available annotation types (differentiated by colour in the original): Topic4 (labels indicative of Topic 4); SentenceTopic4 (Sentences in which Topic4 labels occur); various discourse level information types such as discourse/argument markers and speech acts. Other annotations are available, e.g. sentiment and epistemic. The argumentation engineer can now focus on the close reading of sentences that represent relevant topics, contain the required terminology and argumentational aspects.

For corpus-level exploration and selection, search patterns can be formulated and examined by means of the ANNIC (Annotation in Context) querying and visualization tool in GATE (Aswani et al.(2005)). This tool can index documents not only by content, but also by their annotations and features. It also enables users to formulate versatile queries mixing keywords and information available from any annotation types (e.g. linguistic, conceptual). The result consists of the matching texts in the corpus, displayed within the context of linguistic annotations (not just text, as is customary for KWIC systems). The data is displayed in a GUI, facilitating exploration and the discovery of new patterns.

Searching in the corpus for single annotations returns all those strings that are annotated with the search annotation along with their context and source document. Figure 3 illustrates a more complex query in the top pane by means of which an argumentation engineer wants to explore up to seven token corpus contexts that contain particular term candidates and argument indicators. The query finds all sequences of annotated text where the first string is annotated with *ArgumentationIndicator*, followed by zero to five other *Tokens*, followed by a string with a *TermCandidate* annotation. One of the search results is visualised in the bottom pane by means of query matches and left/right contexts. The coloured bars form an annotation stack that shows the occurrence of selected annotations in the contexts. In this case we see an emphasis argument indicator ”obviously” co-occurring with the term candidates ”Scotland”, ”independent Scotland” and ”choice”.

By inspecting the document more closely the argumentation engineer will be able to produce a structured representation of the identified argument. The ANNIC interface thus uses the annotations to reduce the search space for human engineers, and focuses their attention on passages that are relevant for sourcing arguments. The tool allows incremental refinement of searches, allowing for a interactive way to examine the semantic content of the texts. Also, the argumentation engineer can provide feedback in the form of changing/adding annotations, which will be used in GATE to improve the automated analysis.

5 Related Work

The paper presents developments of an implemented, semi-automatic, interactive text analytic tool that combines rule-based and statistically-oriented approaches. The tool supports analysts in identifying "hot zones" of relevant textual material as well as fine-grained, relevant textual passages; these passages can be used to compose argument graphs in a tool such as DebateGraph. As such, the tool evaluated with respect to user facilitation (i.e. analysts qualitative evaluation of using the tool or not) rather than with respect to *recall* and *precision* (Mitkof(2003)) in comparison to a gold standard. The tool is an advance over graphically-based argument extraction tools that rely on the analysts' unstructured, implicit, non-operationalised knowledge of discourse indicators and content (van Gelder(2007); Rowe and Reed(2008); Liddo and Shum(2010); Bex et al.(2014)). There are a variety of rule-based approaches to argument annotation: (Pallotta and Delmonte(2011)) classify statements according to rhetorical roles using full sentence parsing and semantic translation; (Saint-Dizier(2012)) provides a rule-oriented approach to process specific, highly structured argumentative texts; (Moens et al.(2007)) manually annotates legal texts then constructs a grammar that is tailored to automatically annotated the passages. Such rule-oriented approaches share some generic components with our approach, e.g. discourse indicators, negation indicators. However, they do not exploit a terminological analysis, do not straightforwardly provide for complex annotation querying, and are stand-alone tools that are not integrated with other NLP tools. Importantly, the rule-based approach outlined here could be used to support the creation of gold standard corpora on which statistical models can be trained. Finally, we are not aware of statistical models to extract the fine-grained information that is required for extracting argument elements.

The tool is used to construct or reconstruct arguments in *complex*, *high volume*, *fragmentary*, and *alinearly* presented comments or statements. This is in contrast to many approaches that, by and large, follow the structure of arguments within a particular (large and complex) document, e.g. the BBC's Moral Maze (Bex et al.(2014)), manuals (Saint-Dizier(2012)), and legal texts (Moens et al.(2007)).

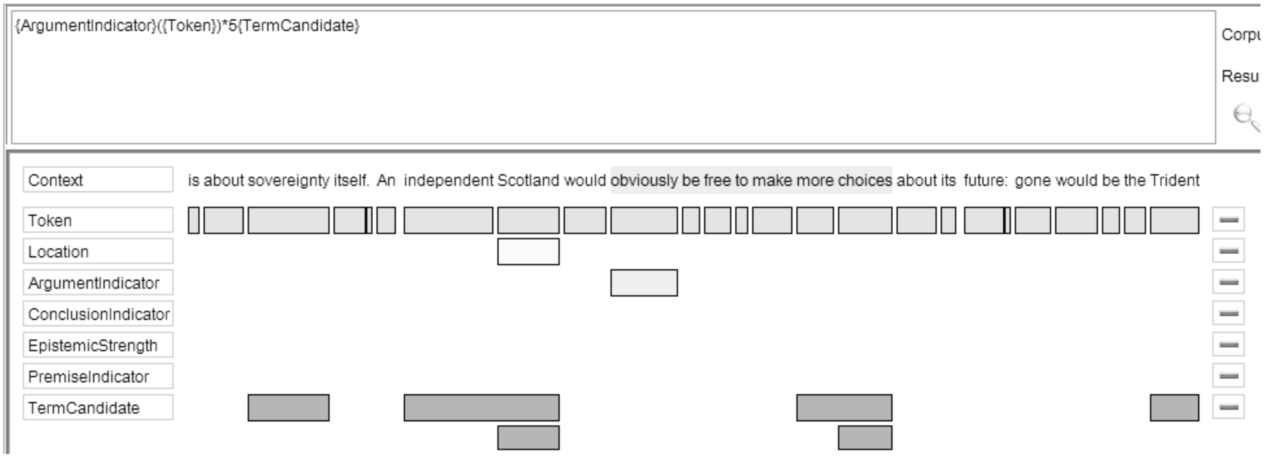
The tool can be modularly developed, adding further argumentation elements, domain models, disambiguating discourse indicators (Webber et al.(2011)), auxilary linguistic indicators, and other parts of speech that distinguish sentence components. More elaborate query patterns could be executed to refine results. In general, the openness and flexibility of the tool provide a platform for future, detailed solutions to issues in argumentation.

6 Discussion

The tool offers a very flexible, useful and meaningful way to query a corpus of text for relevant argument passages, leaving the argument engineer to further analyse and use the results. Having developed in in conjunction with an industrial partner, the next task is to evaluate it with user studies, inquiring whether the tool facilitates or changes the capability to develop arguments for graphs. As a result of this feedback, the tool can be developed further, e.g. adding a summarisation component, automating extraction, augmenting the base terminology (speech acts, propositional attitudes, etc), and creating discourse indicator patterns. The tool can also be used to examine the role of the various components in the overall argument pattern search, investigating the use of, e.g. discourse indicators or speech acts in different discourse contexts.

Acknowledgments

The authors gratefully acknowledge funding from the Semantic Media Network project *Semantic Media: a new paradigm for navigable content for the 21st Century* (EPSRC grant EP/J010375/1). Thanks to Ebuka Ibeke and Georgios Klados for their contributions to the project.



{ArgumentIndicator}({Token})*5{TermCandidate}

is about sovereignty itself. An independent Scotland would obviously be free to make more choices about its future: gone would be the Trident

Figure 3: Searching for Patterns in the Corpus

References

Niraj Aswani, Valentin Tablan, Kalina Bontcheva, and Hamish Cunningham. Indexing and Querying Linguistic Metadata and Document Content. In *Proceedings RANLP 2005*, Borovets, Bulgaria, 2005.

Floris Bex, Mark Snaith, John Lawrence, and Chris Reed. Argublogging: An application for the argument web. *J. Web Sem.*, 25:9–15, 2014.

David Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5):9931022, 2008.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of ACL 2002*, pages 168–175, 2002.

Anna De Liddo and Simon Buckingham Shum. Cohere: A prototype for contested collective intelligence. In *ACM CSCW 2010 - Collective Intelligence In Organizations*, Savannah, Georgia, USA, February 2010.

Diana Maynard, Yaoyong Li, and Wim Peters. NLP techniques for term extraction and ontology population. In *Proceedings of OLP 2008*, pages 107–127, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

Ruslan Mitkof, editor. *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 2003.

Marie-Francine Moens, Erik Boiy, Raquel Mochales-Palau, and Chris Reed. Automatic detection of arguments in legal texts. In *Proceedings ICAIL '07*, pages 225–230, New York, NY, USA, 2007. ACM Press.

Vincenzo Pallotta and Rodolfo Delmonte. Automatic argumentative analysis for interaction mining. *Argument and Computation*, 2(2-3):77–106, 2011.

Glenn Rowe and Chris Reed. Argument diagramming: The Araucaria Project. In Alexandra Okada, Simon Buckingham Shum, and Tony Sherborne, editors, *Knowledge Cartography: Software Tools and Mapping Techniques*, pages 163–181. Springer, 2008.

Patrick Saint-Dizier. Processing natural language arguments with the <TextCoop> platform. *Argument & Computation*, 3(1):49–82, 2012.

Tim van Gelder. The rationale for Rationale. *Law, Probability and Risk*, 6(1-4):23–42, 2007.

Bonnie Webber, Markus Egg, and Valia Kordoni. Discourse structure and language technology. *Natural Language Engineering*, December 2011.

Adam Wyner. Mining fine-grained argument elements. In Elena Cabrio, Serena Villata, and Adam Wyner, editors, *Proceedings of ArgNLP2014*, volume 1341, Bertinoro, Italy, July 2015. CEUR Workshop Proceedings.

Adam Wyner and Wim Peters. On rule extraction from regulations. In Katie Atkinson, editor, *Proceedings of JURIX 2011*, pages 113–122. IOS Press, 2011.

Adam Wyner, Jodi Schneider, Katie Atkinson, and Trevor Bench-Capon. Semi-automated argumentative analysis of online product reviews. In *Proceedings of COMMA 2012)*, pages 43–50. IOS Press, 2012.