Automatic Extraction of Reasoning Chains from Textual Reports

Gleb Sizov and Pinar Öztürk
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
{sizov, pinar}@idi.ntnu.no

Abstract

Many organizations possess large collections of textual reports that document how a problem is solved or analysed, e.g. medical patient records, industrial accident reports, lawsuit records and investigation reports. Effective use of expert knowledge contained in these reports may greatly increase productivity of the organization. In this article, we propose a method for automatic extraction of reasoning chains that contain information used by the author of a report to analyse the problem at hand. For this purpose, we developed a graph-based text representation that makes the relations between textual units explicit. This representation is acquired automatically from a report using natural language processing tools including syntactic and discourse parsers. When applied to aviation investigation reports, our method generates reasoning chains that reveal the connection between initial information about the aircraft incident and its causes.

1 Introduction

Success of an organization is highly depend on its knowledge which is generated and accumulated by its employees over years. However, unless made explicit and shareable, organizations have the risk of losing this knowledge because employees may change jobs at any time, or retire. It is common to document such experience, also for evidence purpose in case of legal problems and governmental regulations. Consequently, many companies and institutions have large collections of textual reports documenting their organizational experience on a particular task, a client or a problem. Industrial incident reports, law suit reports, electronic patient records and investigation reports are the most intuitive examples. The effective use of the knowledge contained in these reports can save substantial time and resources. For example, incident reports can be used to identify possible risks and prevent future incidents, law suit reports constitute precedences for future cases, and patient records might help to diagnose and find an appropriate treatment for a patient with similar symptoms.

Existing search engines are effective at finding relevant documents. However, after retrieval, interpretation and reasoning with knowledge contained in these documents is still done manually with no computer assistance other than basic keyword-based search. In our research, we are aiming to develop methods that will assist users in interpretation and reasoning with knowledge contained in textual reports. The rationale behind our approach is that experts' line of reasoning for understanding and solving a problem can be reused for the analysis of a similar problem. Reasoning knowledge can be extracted from a report by analysing its syntactic and rhetorical structure. When extracted and represented in a computer-friendly way, this knowledge can be used for automatic and computer-assisted reasoning.

In this article, we propose a method for automatic extraction of reasoning chains from textual reports. A reasoning chain is defined as a sequence of transitions from one piece of information to another starting from the *problem description* and leading to its *solution*. Our model is based on a novel

Proceedings of the TextGraphs-8 Workshop, pages 61–69,
Seattle, Washington, USA, 18 October 2013.

graph-based text representation, called Text Reasoning Network (TRN), which decomposes a document into text units, discovers the connections between these text units and makes them explicit. TRN is acquired automatically from text using natural language processing tools including a syntactic parser, a discourse parser and a semantic similarity measure.

We tested our method on aviation investigation reports from Transportation Board of Canada. These reports are produced as a result of investigation of aircraft incidents where experts are assigned the task of analysing an incident and writing down their understanding of what and why it happened. Reasoning chains extracted from the investigation reports reveal the connection between initial information about the incident and its causes. When visualized, this connection can be interpreted and analysed.

The rest of the paper is organized as follows. Section 2 provides an overview of the related research. In section 3, TRN representation is described. Generation of reasoning chains from aviation investigation reports is explained in section 4. Interesting examples of reasoning chains generated by our system are demonstrated and analysed in section 5. In section 6, we discuss the results and elaborate on future work.

2 Related Work

To our knowledge, automatic extraction of reasoning chains from text has not been attempted before. However, we were able to find several papers dealing with text processing tasks relevant to our goal that make use of graph-based representations.

The work done by Pechsiri and Piriyakul (2010) is focused on extraction of causal relations from text and construction of an explanation graph. The relations are extracted between clauses based on mined cause-effect verb pairs, e.g. "If the [aphids infest rice pants], [the leaves will become yellow]." with cause verb "infest" and effect verb "become". The explanation graph is constructed directly from the extracted relations, which is different from our approach where reasoning chains are extracted as paths from the graph-based representation of a report. There is only one example of the explanation graph presented in the paper. This graph is generated from plant disease technical papers capturing part of the domain knowledge. Manual inspection of the graph revealed few mistakes.

An interesting research was conducted by Berant (2012) for his PhD thesis. Unlike Pechsiri and Piriyakul (2010), his approach relies on textual entailment instead of causal relations. Entailment relations are obtained between propositional patterns, e.g. $(X \xleftarrow{subj} desire \xrightarrow{obj} Y, X \xleftarrow{subj} want \xrightarrow{obj} Y)$, using a classifier trained on distributional similarity features. The focus of their work is to exploit transitive nature of entailment relations in learning of entailment graphs. As an application, the authors developed a novel text exploration tool, where a user can drill down/up from one statement to another through the entailment graph. Entailment relations alone, i.e. $text \xrightarrow{entails} hypothesis$, are not sufficient for extraction of reasoning chains because the *hypothesis* often contains the information which is already present in the *text*, making it impossible to create a path from the problem description to the solution. However, when combined with other types of relations they might be useful for out task.

In the paper by Jin and Srihari (2007), authors generate and evaluate evidence trails between concepts across documents. An evidence trail is a path connecting two concepts in a graph where nodes are concepts that correspond to named entities and noun phrases participating in subject-verb-object constructs. Three variations of the representation are tested, each with edges capturing different types of information. In the first one, edges capture word order in text. The second one captures co-occurrence of concepts. The third variation contains edges with weights corresponding to the similarity between contexts of the concepts. Vector space model is used to represent and measure the similarity between the contexts. The concept-based representation is substantially different from TRN but the idea of finding a shortest path between nodes and use it as the evidence is similar. There is one example of the evidence trail shown in the paper: "bush - afghanistan - qaeda - bin ladin", which reveals the connection between topics rather than concrete pieces of information.

A graph-based representation similar to (Jin and Srihari, 2007) have been applied for Textual Case-

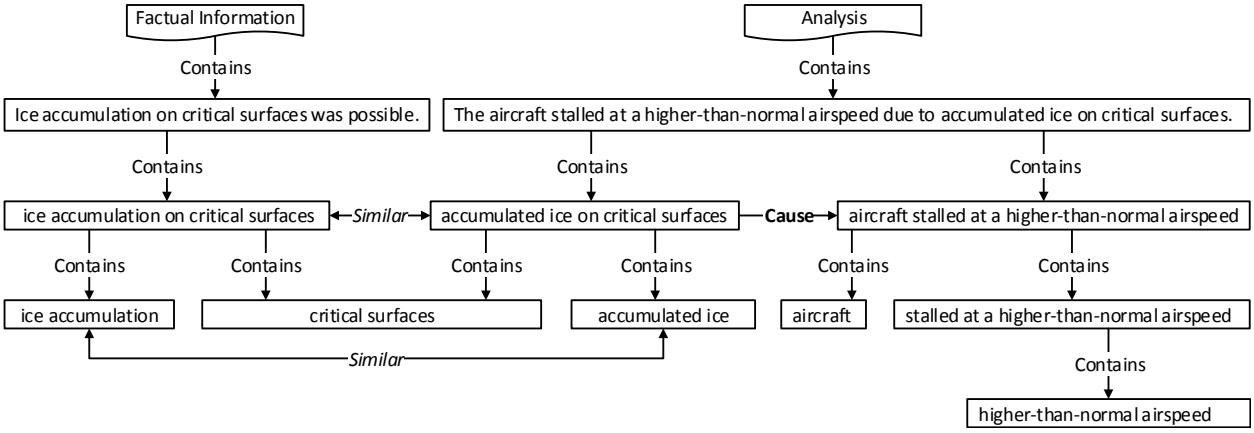

Figure 1: Two sentences represented as Text Reasoning Network.

Based Reasoning (TCBR) (Lenz and Burkhard, 1996; Cunningham et al., 2004), a task of automatically solving a new problem given a collection of reports describing previous problems with solutions. The dataset we use in our research can be considered a TCBR dataset, since each report contains a problem description and a solution part. Problem-solving based on knowledge represented in textual form is a tough task and in practice TCBR approaches either do classification of a problem into predefined classes or retrieve a report that describes a problem similar to a query problem. In the later case, information retrieval methods are utilized, including graph-based representations for computing similarity between documents as it is done by Cunningham et al. (2004). Their representation, inspired by Schenker et al. (2003), contains terms as nodes and edges connecting the adjacent terms in text. Nodes and edges are labelled with the frequency of their appearance and with the section where they appear, i.e. title or text. Infrequent terms are removed. In addition, domain knowledge is introduced as a list of important domain terms that are preserved even if their frequency is low. The similarity used is based on maximum common sub-graph. When tested on summary documents from a law firm handling insurance cases, the results show improvement over vector space model representations.

3 Text Reasoning Network

In our approach, a reasoning chain is extracted as a path from the graph-based text representation. An appropriate representation is crucial because chains extracted from it are only as good as the representation itself. In this section we introduce a novel graph-based text representation, called Text Reasoning Network (TRN), which is a graph with two types of nodes: text nodes and section nodes; and three types of edges: structural, similarity and causal. Figure 1 shows two sentences from a report represented as TRN with section nodes on the top and all the text nodes below them. The representation is acquired automatically from text by the following procedure: (1) syntax trees obtained from a syntactic parser are added to the graph, (2) section nodes are attached to sentence nodes, (3) similarity edges are added between similar text nodes, (4) cause relations identified by a discourse parser are added. The rest of this section provides details on the structure of TRN and methods used to generate it from text.

3.1 Nodes and Structural Relations

We are aiming to extract chains that capture the information used by the author of a report to reason about the problem at hand. Graph-based text representations described in section 2 use individual terms or short phrases as nodes. Small text units such as these are unable to capture sufficient information for our purpose. Another popular choice for a node in a textual graph is a sentence, which captures a more or less complete piece of information and is easy to interpret. However, a complex sentence may contain several pieces of information where only one is used in a reasoning chain.

A syntax tree provides a natural decomposition of

a sentence into its constituents. Since it is hard to determine beforehand the size of constituents that would be useful in a reasoning chain, we decided to incorporate all the S (sentence, clause), NP (noun phrase) and VP (verb phrase) nodes from syntax trees produced by Stanford Parser (Klein and Manning, 2003). These nodes are referred to as *text nodes*. In addition to text nodes, the structure of a syntax tree is also retained by adding *structural relations Contains* and *PartOf* to TRN that correspond to relations between parent and children text units in the syntax tree. Figure 1 shows text nodes extracted from two sentences along with *Contains* relations between them. *PartOf* edges are not shown to avoid the clutter.

Graphs extracted from different sentences in a document are combined into one. Each node has a unique identity that is composed of a sequence of stemmed words with stopwords removed. The major implication of this is that if two sentences overlap, they will share one or several nodes, e.g. node "critical surfaces" in figure 1.

In addition to text nodes, there are also *section nodes* corresponding to parts of a document, e.g. "Factual Information" and "Analysis" nodes in figure 1. These nodes capture the structure of a document. Text nodes containing a complete sentence, also referred to as *sentence nodes*, are attached to section nodes by structural relations.

3.2 Similarity Relations

In addition to structural relations, text nodes are connected through similarity relations. To obtain these relations, a similarity value is computed for each pair of text nodes of the same category (S, VP, NP) that are not in the same sentence. *Similar* edges are added to the graph for node pairs with the similarity value above a predefined threshold, e.g. nodes "ice accumulation" and "accumulated ice" in figure 1.

Our similarity measure finds one-to-one alignment of words from two text units to maximize the total similarity between them. For words we compute *LCH* (Leacock et al., 1998) similarity, based on a shortest path between the corresponding senses in WordNet. A complete bipartite graph is constructed and the maximum weighted bipartite matching is computed using the Hungarian Algorithm (Kuhn, 1955). Nodes in this bipartite graph represent words from the text units while edges have weights that correspond to similarities between words. Maximum weighted bipartite matching finds a one-to-one alignment that maximizes the sum of similarities between aligned words. This sum is normalized to lie between 0 and 1 and is used as the final value for the similarity between text units. If the value is higher or equal 0.6 a *Similar* edge is added between the corresponding nodes.

3.3 Causal Relations

Causal relations are essential for analysis and decision making allowing inference about past and future events (Garcia-Retamero et al., 2007). As seen in (Pechsiri and Piriyakul, 2010) causal graphs extracted from domain-specific documents provide a powerful representation of expert knowledge.

State-of-the-art techniques for extraction of causal relations from text use automatic classifiers trained on lexical features to recognize relations between subject and object in a clause or between verbs of different clauses (Chang and Choi, 2005; Bethard and Martin, 2008). Causal relations are among discourse relations defined by Rhetorical Structure Theory (Mann and Thompson, 1988). Therefore, a discourse parser can be used to obtain them from text. The advantage of this approach is that a discourse parser recognizes relations between larger text units. Few discourse parsers are available that can parse an entire document. For our work we used PDTB-Styled End-to-End Discourse Parser by Lin et al. (2010), which makes use of machine learning techniques trained on Penn Discourse Treebank (PDTB) (Prasad et al., 2008). Cause relations identified by the parser are added to TRN graph by mapping arguments of the relations to text nodes and then adding *Cause* edges between them.

4 Generation of Reasoning Chains from Aviation Accident Reports

Generation of a reasoning chain from a report is a three-stage process: (1) a report is converted from text to a TRN graph, (2) given a start and an end node, several paths are extracted from the graph, (3) paths are combined, post-processed and visualized.

4.1 Dataset

In our work we use aviation investigation reports from Transportation Safety Board of Canada[1]. Each report in this collection documents an aircraft incident and contains the following sections: (1) "Summary" is a brief description of the incident, (2) "Factual Information" (further referred to as "Factual") contains details about the aircraft, pilot, weather conditions, terrain and communication with controllers (3) "Analysis" is a discussion of the incident with the purpose to explain it based on the information presented in the previous section, (4) "Findings as to Causes and Contributing Factor" (further referred to as "Causes") is a brief enumeration of findings that most likely caused the incident.

The reports were downloaded from Transportation Board of Canada website as html documents. Text and structure were extracted from html using a custom Java component developed based on manual analysis of the html source. Preprocessing steps including tokenization, sentence splitting and part-of-speech tagging were accomplished using ANNIE components in GATE NLP platform (Cunningham et al., 2002).

4.2 Extraction of Reasoning Chains

We define a reasoning chain as the shortest path through a TRN representation of a report starting from a sentence in "Summary" and ending at one of the sentences in "Causes" section. The rationale behind this decision is to reveal the author's reasoning line starting from the initial information about the incident contained in "Summary" and leading to incident causes in "Causes" section. Hence, the path finding process is constrained to follow the direction from "Summary" to "Causes" through "Factual" and "Analysis" sections. The reasoning chain path with constraints is defined by the following context-free grammar in Backus-Naur Form (optional items in [...]):

⟨*path*⟩ ::= ⟨*summary-path*⟩ [⟨*edge*⟩ ⟨*factual-path*⟩] [⟨*edge*⟩ ⟨*analysis-path*⟩] ⟨*edge*⟩ ⟨*causes-path*⟩

⟨*summary-path*⟩ ::= ⟨*summary-node*⟩
| ⟨*summary-node*⟩ ⟨*contains-edge*⟩ ⟨*summary-path*⟩

⟨*factual-path*⟩ ::= ⟨*factual-node*⟩
| ⟨*factual-node*⟩ ⟨*edge*⟩ ⟨*factual-path*⟩

⟨*analysis-path*⟩ ::= ⟨*analysis-node*⟩
| ⟨*analysis-node*⟩ ⟨*edge*⟩ ⟨*analysis-path*⟩

⟨*causes-path*⟩ ::= ⟨*causes-node*⟩
| ⟨*causes-node*⟩ ⟨*partof-edge*⟩ ⟨*causes-path*⟩

⟨*edge*⟩ ::= ⟨*partof-edge*⟩
| ⟨*contains-edge*⟩
| ⟨*similar-edge*⟩
| ⟨*cause-edge*⟩

Several paths are obtained for each "Summary" sentence, each of which starting at one of the text nodes contained in the sentence. These paths are then combined into a reasoning graph. Before visualization, a post-processing algorithm is applied to make the reasoning graph more compact. The algorithm collapses a sequence of structural edges of the same type into a single edge, e.g. $A \xrightarrow{contains} B \xrightarrow{contains} C$ is converted into $A \xrightarrow{contains} C$ if there is no other edge attached to B. The compressed graph (shown in figures 2, 3 and 4) is visualized using JGraphX library with hierarchical layout for automatic positioning of nodes.

5 Examples and Analysis

In this section we analyse three reasoning graphs generated by our system. These graphs were selected mainly because of their compact size and ease of interpretation even for someone who is not an aviation expert. Every chain starts with a sentence from "Summary" on the top of the figure and ends with one or several sentences in "Causes" on the bottom. For each node a contained text unit is displayed, followed by one or several letters in parenthesis indicating which section of the report this text node is observed in: S - "Summary", F - "Factual", A - "Analysis", C - "Causes".

Figure 2 shows the reasoning graph with one branch expressing that the captain's focus on "setting climb power" and the "landing gear" prevented him from paying attention to the "aircraft altitude" so the "sink rate was undetected and aircraft struck the ground". The start and the end sentences are not similar and it is the sentence from the "Analysis" section that connects these two.

[1]Aviation Investigation Reports are available at `http://goo.gl/k9mMV`

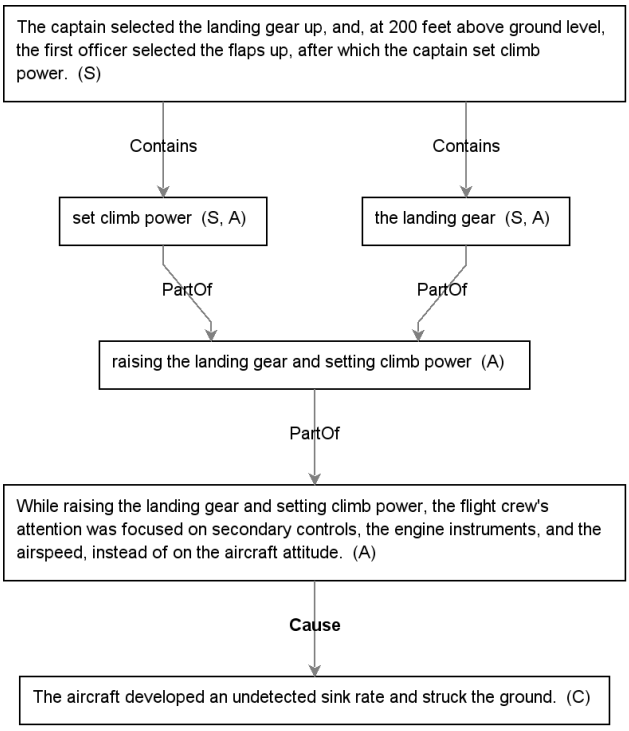

Figure 2: A reasoning graph from report A05O0225 (available at `goo.gl/SZpTS`)

The graph in figure 3 has two branches. The left branch directly points to a sentence with "the auxiliary fuel pump" but it does not explain "a poor electrical connection". The right branch, however, is longer and goes from "switched fuel tank" to "fuel flow" and then to "fuel pressure", which is part of a sentence in the "Cause" section that includes this text segment: "reduction of fuel pressure, preventing normal engine operation".

The graph in figure 4 contains two branches as well. The left branch picks up the location of the flight "Deer Lake" which relates to "icing conditions" although the text node suggesting "a lower altitude was requested to remain clear of icing conditions" makes this branch incoherent. The right branch provides a connection between "Provincial Airlines Limited" and "no requirement" in their "standard operating procedures" for a "method for ensuring the correct selection of AFCS climb modes". The chain goes through "an inappropriate AFCS mode" providing a good idea of the incident cause.

Reasoning chains extracted by the system provide a brief overview of the authors' reasoning line showing how a basic information about the incident is connected to its causes. However, some chains are less informative than others (left branch in figure 3) or incoherent (left branch of 4). In the former case the chain could be made more informative if the system will be queried to find evidence for "poor electrical connection" in addition to "the auxiliary fuel pump". In the latter case, the chain becomes incoherent because "icing conditions" is used in different contexts where the first sentence states the lack of "icing conditions" and the second the presence of "icing conditions". It is possible to account for this inconsistency by introducing a preference for larger text units capturing more context or by recognizing negations/absence.

6 Conclusion and Future Work

This paper presents a method for extraction of reasoning chains from textual reports. The method is based on a graph-based text representation that captures both the structure and the content of the report. Extracted reasoning chains provide a convenient way to visualize information used by a domain expert to reason about causes of an aircraft incident. It may help in analysis of future incidents and opens the possibility for automatic or computer-assisted analysis. The methods can be adapted to other domains and applications by defining appropriate start nodes, end nodes and constraints like it is done in section 4.2.

Extraction of reasoning chains is a new task and there are yet no evaluation measures available. One of the primary goals for our future work is to develop a formal evaluation procedure for this task. An intrinsic evaluation will require manually constructed reasoning chains as the gold standard to compare the automatically extracted ones with. For the extrinsic evaluation, reasoning chains can be used in TCBR task for solution retrieval and evaluated with TCBR evaluation measures (Raghunandan et al., 2008; Adeyanju et al., 2010). We also plan to continue our work on the representation by adding new types of relations to TRN and on the reasoning chain extraction algorithm by adapting flow networks instead of shortest path for extraction of reasoning chains.

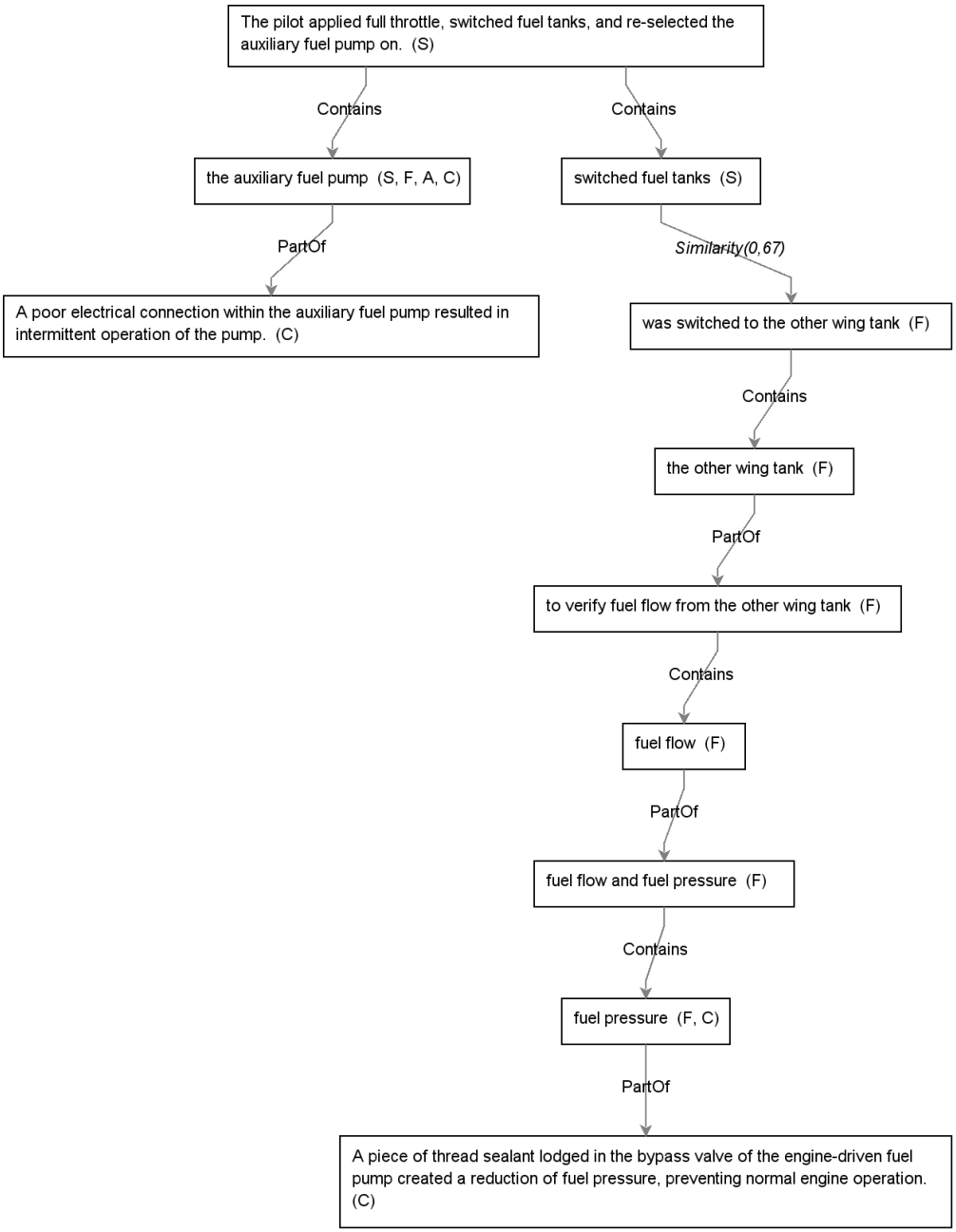

Figure 3: A reasoning graph from report A05O0146 (available at `goo.gl/MPMIq`)

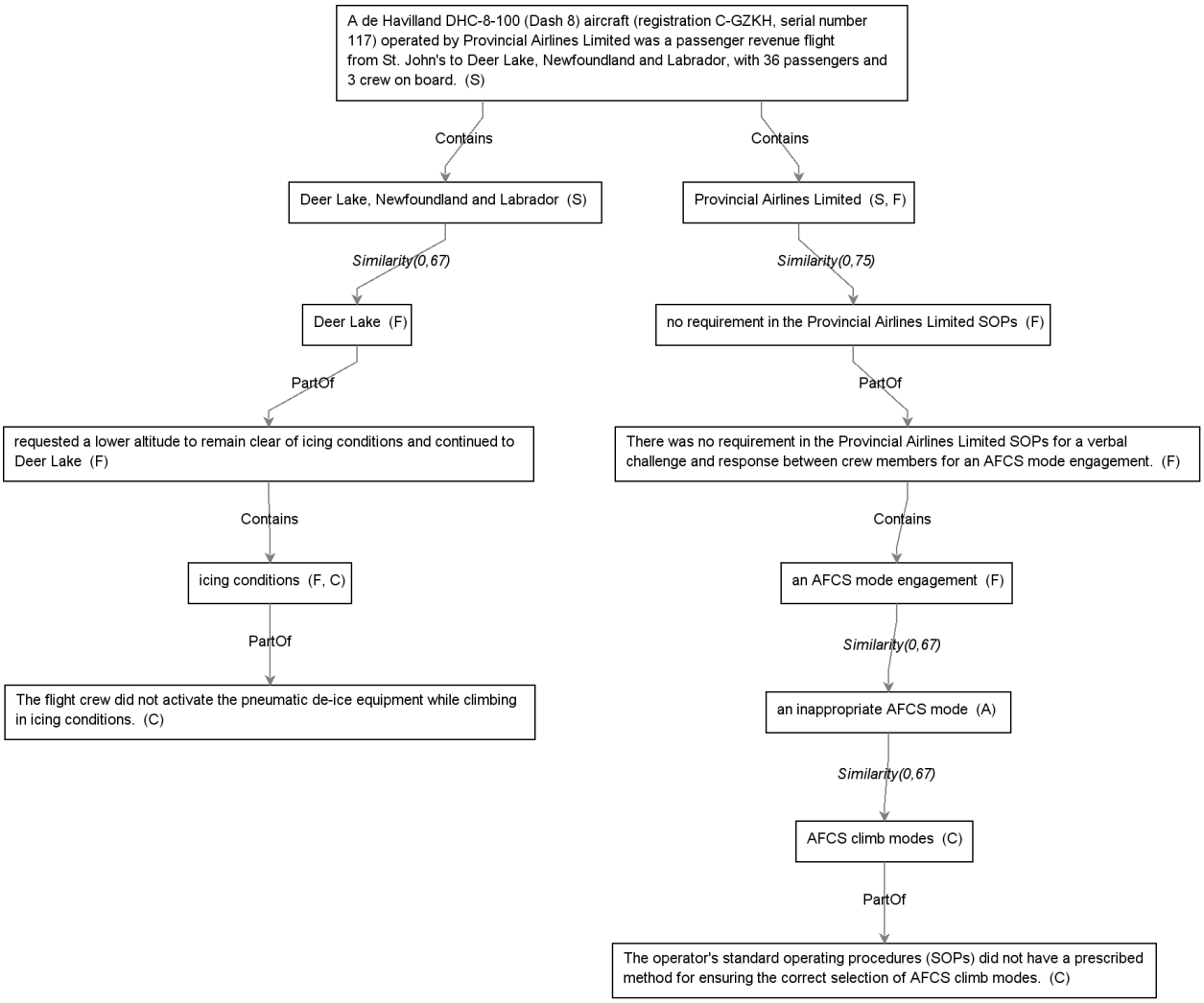

Figure 4: A reasoning graph from report A05A0059 (available at `goo.gl/u1CXI`)

References

Ibrahim Adeyanju, Nirmalie Wiratunga, Robert Lothian, and Susan Craw. 2010. Applying machine translation evaluation techniques to textual cbr. In *Case-Based Reasoning. Research and Development*, pages 21–35. Springer.

Jonathan Berant. 2012. *Global Learning of Textual Entailment Graphs*. Ph.D. thesis, Tel Aviv University.

Steven Bethard and James H Martin. 2008. Learning semantic links from a corpus of parallel temporal and causal relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 177–180. Association for Computational Linguistics.

Du-Seong Chang and Key-Sun Choi. 2005. Causal relation extraction using cue phrase and lexical pair probabilities. In *Natural Language Processing–IJCNLP 2004*, pages 61–70. Springer.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. Gate: an architecture for development of robust hlt applications. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 168–175, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Colleen Cunningham, Rosina Weber, Jason M Proctor, Caleb Fowler, and Michael Murphy. 2004. Investigating graphs in textual case-based reasoning. In *Advances in Case-Based Reasoning*, pages 573–586. Springer.

Rocio Garcia-Retamero, Annika Wallin, and Anja Dieckmann. 2007. Does causal knowledge help us be faster and more frugal in our decisions? *Memory & cognition*, 35(6):1399–1409.

Wei Jin and Rohini K. Srihari. 2007. Graph-based text representation and knowledge discovery. *Proceedings of the 2007 ACM symposium on Applied computing - SAC '07*, page 807.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Claudia Leacock, George A Miller, and Martin Chodorow. 1998. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

Mario Lenz and Hans-Dieter Burkhard. 1996. Case retrieval nets: Basic ideas and extensions. In *KI-96: Advances in Artificial Intelligence*, pages 227–239. Springer.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2010. A pdtb-styled end-to-end discourse parser. Technical report, Cambridge Univ Press.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Chaveevan Pechsiri and Rapepun Piriyakul. 2010. Explanation knowledge graph construction through causality extraction from texts. *Journal of Computer Science and Technology*, 25(5):1055–1070.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

M. A. Raghunandan, Nirmalie Wiratunga, Sutanu Chakraborti, Stewart Massie, and Deepak Khemani. 2008. Evaluation measures for tcbr systems. In *Advances in Case-Based Reasoning*, pages 444–458. Springer.

A. Schenker, M. Last, H. Bunke, and A. Kandel. 2003. Clustering of web documents using a graph model. *Web Document Analysis: Challenges and Opportunities*, pages 1–16.