To appear in the proceedings of the Workshop on Textual Case-Based Reasoning: Beyond Retrieval, ICCBR-07. Belfast, Northern Ireland.

Conversation for Textual Case-Based Reasoning

Kalyan Moy Gupta[1] and David W. Aha[2]

[1]Knexus Research Corp.; Springfield, VA 22153; USA
[2]Naval Research Laboratory (Code 5514); Washington, DC 20375; USA
[1]kalyan.gupta@knexusresearch.com
[2]david.aha@nrl.navy.mil

Abstract. Conversational Case-Based Reasoning (CBR) systems assist their users in formulating queries for case retrieval. Existing textual CBR (TCBR) systems support conversation using only hand-crafted features and indices. However, to be practical, TCBR systems that require conversation should automatically generate their features. This is a difficult problem because TCBR applications routinely involve thousands of interrelated features. In this paper, we explore candidate methodologies to address this problem. We describe domain and human factors issues that must be considered for TCBR conversation. With effective and efficient conversation as our goal, we propose a method for conducting conversation with automatically generated feature vocabularies. We illustrate our approach with examples from a database of air investigation reports and identify underlying problems.

1 Introduction

Textual case-based reasoning (TCBR) is a case-based reasoning methodology that predominantly employs cases derived from artifacts containing blocks of *unrestricted natural language text* such as incident reports, legal briefs, and email communications. In contrast, most other case-based reasoning (CBR) methods use structured data in which the non-numeric attributes have restricted canonical text as values such as product categories, and movie genre.

CBR methods can be distinguished by whether they involve *conversation* for incremental query formulation (Aha *et al*., 2001; Gupta & Aha, 2003). Many problem-solving tasks do not require query formulation or automate the query formulation process without user interaction. For example, email spam filtering and message classification do not require conversation and can be fully automated by TCBR. However, decision support applications of CBR such as product recommendation and industrial troubleshooting are typically designed to engage users in conversations.

Existing TCBR systems that support conversation require manually engineered cases derived from the original source documents such as trouble shooting records and incident reports (e.g., Aha, 1998; Gupta *et al*., 2002). Case engineering includes steps such as feature vocabulary identification, feature organization, case indexing, and solution preparation. It produces a small subset of features from the entire case vocabulary of roughly 10,000-15,000 terms. These features are sometimes organized into hierarchies and non-hierarchical structures (e.g., Gupta *et al*., 2002; Gu & Aamodt, 2005). These organized feature vocabularies make existing conversation approaches feasible and effective.

Unfortunately, manual case engineering is expensive, time consuming, and potentially impractical for short-lived dynamic applications. Although some recent TCBR research has focused on automatically developing and organizing features (e.g., Massie *et al.,* 2007), the performance of an automatically generated feature vocabulary is likely to be inferior to a manually generated vocabulary when it is used for conversation. For example, it is difficult to automatically construct list-valued attributes and automatically detect high-fidelity taxonomic and causal relations (Gupta *et al*., 2004). In this paper, we argue that new techniques for conducting conversation are needed that operate on *automatically developed* feature vocabularies and case indices in TCBR systems. We develop a framework for conducting conversations and identify candidate methodologies that can be used to implement such a framework.

In the rest of this paper, we first introduce the air investigation reports domain we for illustrating our methods, then review the case indexing process and examine the differences between manually and automatically developed feature vocabularies. Afterwards, we take a fresh look at the role of conversation in CBR systems and TCBR systems in particular, and methods to assess conversation performance. We then use these measures to frame the space of methodologies for conversation generation. We illustrate the use of such a methodology on a problem involving air investigation reports, and conclude with suggested directions for future research.

2 Air Investigation Reports

We illustrate our methodologies and concepts using the domain of air investigation reports obtained from the Canadian Transportation Safety Board (TSB, 2007). We analyzed 7 reports that record air incidents that occurred in 2005. Our analysis of the entire text contents of these reports, ignoring the report structure, showed that they contain over 53,000 terms and have a vocabulary of over 7,500 unique terms. By considering only the terms that occur at least four times in the corpus, we reduced the vocabulary to 1640 terms (1000 nouns, 360 verbs, 200 adjectives, 80 adverbs), which is still a large vocabulary. We then manually selected 25 terms for the feature vocabulary (see Figure 1), focusing on the most frequent nouns and verbs.

Feature Vocabulary (Terms)

```
aerodrome, aircraft, cloud, collide, crash, crew,
destroy, drown, fire, fog, ground, helicopter, ice,
lake, landing, night, passenger, pilot, rotor,
rudder, snow, submerge, survive, tail, take-off
```

Figure 1: Example feature vocabulary manually drawn from 17 air investigation reports

3 Manual vs. Automatic Case Indexing

TCBR systems routinely use a “bag-of-words” (terms as binary-valued attributes) representation for cases; a representation we focus on in this paper. Case indexing assigns a set of attribute-value pairs (i.e., features) to cases, where the features are selected from a feature vocabulary. For example, a case indexing process for a TCBR method might build a vocabulary by selecting a subset of unique terms from the cases as features and assigning a value of 1 to the cases that include a given term and 0 otherwise. Vocabulary development usually includes the following steps:

1. *Identify attributes*: The first step is to identify a unique set of terms and/or phrases from the source documents. This can be performed manually, automatically, or semi-automatically. It usually produces either binary or numeric-valued terms/phrases as attributes. For example, our air incident reports domain, although it includes only 17 documents, has 7500 unique terms that could be used as features.
2. *Identify values*: For nominal-valued attributes, its set of possible values must be identified. For example, the attribute `aircraft` in our example domain can have the nominal values `Cessna`, `Seaplane`, and `C130 Transport`. Values are identified using a predominantly manual process and existing TCBR research has not focused on developing or applying automated techniques to this task. Existing automatic methods typically disregard the relationship between the terms `aircraft` and `Cessna` and instead create two binary-valued attributes. Knowledge resources (e.g., domain ontologies) could help to automate this step.
3. *Select attributes*: It is impractical to use the complete case vocabulary as features. For example, using 7500 features to discriminate among 17 sources in our example domain is inappropriate. Therefore, the feature vocabulary must be reduced. When this task is performed manually, knowledge engineers select a small set of salient terms/phrases. However, if they lack suitable tools, they may not consider the ability of these terms to distinguish the solution categories. If features are selected automatically for use in conversation using existing feature selection methods, we conjecture that they are likely to be of inferior quality than a set of manually selected features. The reason is that knowledge engineers utilize significant domain knowledge for accurate identification of features, which is inaccessible to knowledge poor automatic feature selection methods.
4. *Organize attributes:* A prevalent problem in TCBR applications is feature correlation, which adversely affects similarity assessment and case retrieval performance. Feature correlation occurs because document authors paraphrase and redundantly use words to express the same concept. Linguistic relationships among words (e.g., synonymy, hyponymy (is-a-type-of), and meronymy (is-a-part-of) among terms) are the main cause of attribute correlation. Therefore, some TCBR systems attempt to organize attributes into taxonomies and causal networks to enable feature value inferencing and more efficient query formulation (e.g., Brüninghaus & Ashley, 2005; Gupta *et al.,* 2002; Gu & Aamodt, 2005). Attribute organization has usually been performed manually. However, some automatic methods learn attribute-value inference rules that tolerate correlation (e.g., Gupta *et al.*, 2004; Massie *et al.*, 2007). Novel conversation methods are needed that can use such rules and relations.
5. *Construct attributes*: During manual feature vocabulary development, knowledge engineers often introduce new terms and phrases that do not occur in the source document. Attribute construction is performed to simplify the retrieval task by more clearly distinguishing solutions, and to reduce the feature vocabulary size. Some automatic methods such as Latent Semantic Indexing (Deerwester *et al.,* 1989) construct composite attributes that combine terms to reduce feature

correlation. However, no methods for using automatically constructed features in TCBR conversations have been developed.

Once an initial feature vocabulary has been developed, it can be used to assign features to documents. Some application domains, such as those involving legal reasoning, may have established feature vocabularies that can be directly used to represent cases. Manually representing cases requires reading the source documents and assigning the appropriate features to them. Automatic indexing methods include calculating term occurrence in a document or indexing it using learned classifiers (e.g., Brüninghaus & Ashley, 2005). Next, we review conversation and its measures and use these measures to develop a framework for TCBR conversation.

4 Conversation

Conversation in a CBR system is a cooperative exchange between the system and its user to formulate queries for effective case retrieval and problem solving. Consider the query formulation process for a variety of problem-solving tasks. For a diagnosis task, it involves observing and specifying symptoms and test results, while for legal reasoning it may involve describing the charges and arguments for and against the defendant. For a planning task, query formulation may focus on specifying characteristics of the initial and goal states (i.e., situation descriptions). Conversation typically consists of a user identifying an initial set of features, followed by an iterative process in which the system prompts the user with a set of additional features to consider (Gupta & Aha, 2003), from which the user selects one or more. In each iteration the query is edited. For example, Conversational CBR (CCBR) systems used for customer support applications synthesize a list of attributes and values from the retrieved cases and present them to the user (Aha *et al*., 2001).

4.1 Domain and Human-Factors Issues in Conversation

The following issues must be considered when designing conversation algorithms:

1. *Problem domain complexity*: The size of the space of possible queries can be used as a measure of problem domain complexity. In TCBR, in the worst case, the space is combinatorial in the size of the feature vocabulary. Because TCBR system vocabularies are large (≥ 1000), finding the (potentially unique) subset of features for indexing each case is difficult. Conversation helps to manage this complexity by presenting the user with features that distinguish the known cases. Although users can formulate queries without conversation, without significant domain expertise, the queries are likely to be incomplete and/or ill-formed resulting in poor or failed case retrieval.
2. *Cognitive limitations:* A user may be unable to recall and identify relevant features for query formulation. For most common cognitive tasks, human recall performance can be dramatically improved when the task is transformed into a recognition task (Anderson, 1990). For example, subjects perform better on multiple choice question-answer tests (a recognition task) than on open-ended questions (a recall task). Similarly, prompting during conversation transforms

query formulation into a recognition task. It *reminds* the user of features that s/he knows about but was unable to recall without prompting.

3. *Human-system terminology gap*: Compared to humans, TCBR systems only recognize and process a limited vocabulary. Thus, a user's vocabulary can differ considerably from that of the system's vocabulary. For example, the system and user may refer to the same concept with different but synonymous terms. Prompting during conversation helps users to select the vocabulary known to the system, so that the user can convey which indices to use for retrieval.
4. *Human-system knowledge gap*: The user and the system may be knowledgeable about different parts of the domain. Depending on a user's personal experience, this knowledge gap can be large. For example, users of self-help or customer support applications typically are not domain experts and are thus not technically articulate in the domain. However, problem reports are authored by experts, who are typically more knowledgeable than end users and are the source of the system's feature vocabulary. A good conversation helps to bridge this gap by reminding, informing, and educating the user. The user does not need to completely understand the relevant concepts and yet, through such conversations, can identify suitable features for describing them to the system.

In summary, conversation improves *query formulation performance* by narrowing the choice of feature combinations, reminding the user, and bridging terminological and domain knowledge gaps. Although efficient and accurate retrieval requires queries with independent features, conversation is facilitated by related terms and features for the reasons stated above.

4.2 Measuring Conversation Performance

To develop measures of conversation performance, we introduce the notion of the *best query*. The best query for a problem is one that helps the user to retrieve the case(s) that have the best solution(s). Therefore, the goal of a conversation is to help the user to formulate a query that is close or identical to the best query. We denote the query formulated by a conversation as the *conversational query* and its retrieval performance as the *conversational-query-retrieval-performance.* Similarly, we denote the retrieval performance resulting from the best query as the *best-query-retrieval-performance*. Given these, we introduce the following measures for assessing conversation performance:

1. *Effectiveness*: A conversation's effectiveness is measured by its ability to formulate a query whose retrieval performance is close or identical to that of the best-query-retrieval-performance. Therefore, conversational effectiveness is measured by the ratio of conversation-query-retrieval-performance and best-query-retrieval-performance. The problem representation of the case(s) with the best solution for a given problem can serve as the best query.
2. *Efficiency*: A conversation's efficiency is measured by the number of interaction operations that the user must perform to formulate their query. This number can be measured by the number of prompts or suggestions that the user must browse and/or select to completely formulate a query.

5 Effective and Efficient TCBR Conversation Methods

To describe suitable conversation methodologies, we introduce two concepts. First, a *prompt* is a user interface element that allows a user to select features. For example, "`aircraft or helicopter?`" is a prompt. A prompt can be associated with one or more features. For example, the prompt "`aircraft or helicopter?`" is associated with two binary features: `aircraft` and `helicopter`. However, the prompt "`rotor?`" is associated with only one feature (see Figure 2).

Prompt-set

```
1. aircraft or helicopter?
2. landing or takeoff?
3. snow and ice?
4. crew, passenger, and pilot?
5. transport and cargo?
6. fire and destroyed?
7. collide and crash?
8. water, submerge, and drown?
9. tail?
10.rotor?
11.drown or survive?
12.rudder?
13. aerodrome?
```

Figure 2: Example prompt-set derived from the feature vocabulary

Second, a *prompt-set* includes all the prompts that are presented to a user prior to case retrieval. During query formulation several prompt-sets may be generated, one prior to each retrieval operation. Figure 2 shows a prompt-set with an ordered list of prompts that includes all the features from our example feature vocabulary.

5.1 Prompt Types and the Conversation Process

In this section, we categorize prompts into types based on the user interaction operations they support. We also suggest some methods for generating these prompt types in Section 5.2 but leave their development and application for future research.

1. *Select one-of-one*: This prompt type is associated with only one feature, which the user can either select or deselect. For example the prompts "`rudder?`" and "`aerodrome?`" in our example domain are each associated with only one feature. This prompt type gives users maximal flexibility in selecting features. However, it provides the least efficient form of interaction. For example, if the vocabulary of the feature set shown in Figure 1 was converted to a prompt-set using only this prompt type, then a maximum of 25 prompts would be required. To select a subset of feature of size k (< 25) would require k user operations. Below, we describe additional prompt types that can enable more efficient conversations.

Independent features	Mutually exclusive features	Co-occurring features
		[snow, ice]
• rudder	{aircraft,helicopter}	[water,submerge,drown]
• aerodrome	• {landing, takeoff}	[crew,passenger,pilot]
• tail	• {drown, survive}	[transport, cargo]
• rotor	• {ground, lake}	[fire, destroyed]
		[collide, crash]

Figure 3: Independence, mutual exclusivity, and co-occurrence among (binary) features

2. *Select one-of-many* (exclusive-or): This applies to prompts associated with multiple features, among which the user can select only one. For example, the prompt "`aircraft or helicopter?`" is associated with two features (`aircraft` and `helicopter` respectively) but allows the user to select only one of them. The features presented to the user are mutually exclusive (i.e., they do not co-occur in a query). Figure 3 shows sets of mutually exclusive features drawn from the example feature vocabulary. This prompt type reduces the number of prompts in a prompt-set and enables more efficient conversations by allowing users to eliminate significant portions of the query space in one conversational iteration.
3. *Select all-of-many* (conjunction): This operation also applies to prompts associated with multiple features. However, unlike select one-of-many prompts, the user selects all the features in the prompt with a single interaction operation. For example, the prompt "`transport and cargo?`" enables a user to select two features using a single operation. This prompt type is the most efficient among the prompt types listed here. However, it provides less flexibility in feature selection as it constrains users to select a combination of features deemed to be appropriate by the system. The features associated with this prompt type co-occur frequently and have a strong co-occurrence relation in queries.
4. *Select some-of-many* (m-of-n): This also applies to prompts associated with multiple features. It relaxes the selection constraint of the select all-of-many prompts from *all* to *some*. This type of prompt is suitable for situations when the co-occurrence relation among features is weak. Figure 3 displays some co-occurring features for this domain. For example, the features *fire* and *destroyed* are highly likely to co-occur in the aircraft incident reporting domain. This prompt type can also improve conversation efficiency by helping users to more easily browse and select multiple features. It supports the browsing task by displaying associations among features that have potentially valid interpretations in the problem domain.

These prompt types are used in an iterative process consisting of three steps:

1. *Prompt-set generation*: On the basis of the features included in a query, the CBR system generates a prompt-set that the user can review and use to select features.
2. *Prompt-set presentation*: Assuming a graphical user interface for presentation, the prompts in a prompt-set can be presented as an ordered list. A list allows a *linear review* and interaction with the prompt (see Figure 2). Prompts can also be hierarchically presented as a tree, for example. Such a presentation would allow a

non-linear user interaction. This can support more efficient prompt browsing and feature selection than linear presentations. However, prompt generation algorithms must also generate the necessary structure and relationships among prompts underlying the presentation.

3. *Prompt-set interaction*: User interaction with a prompt-set depends on the type of a prompt and its presentation. The presentation constrains user browsing behavior and the prompt type constrains a user's selection behaviors.

The prompt-set in Figure 2 is an example of a linear presentation of multiple prompt types.

5.2 Prompt-set Generation Methodologies

Prompt-set generation consists of the following tasks:

1. *Feature selection*: The most important prompt-set characteristic for determining conversation performance is its size, which in turn depends on the number of features to be considered for prompting. Therefore, feature selection is the primary computational task in prompt-set generation. Feature selection methods have used term frequency, information gain (Yang & Pederson, 1997), boosted decision stumps (Wiratunga *et al.*, 2004), and rough sets (Gupta *et al.*, 2006) for this task. Although small sets of independent features are desirable for accurate retrieval, larger sets with redundant and correlated features might be more desirable for reminding and informing the user about the important features for query formulation. Therefore, new feature selection techniques that trade off these competing objectives may be needed.

   Another issue pertinent to TCBR feature selection is the lack of solution classes. Therefore, feature selection methods for unsupervised learning tasks are also needed (e.g., those that use clustering to generate solution classes and then use conventional feature selection techniques for supervised learning tasks).

2. *Feature interrelationship identification and prompt composition*: To compose prompts, feature interrelationships such as mutual exclusivity and co-occurrence should be identified. Various techniques for relation discovery and/or association rule mining can be used for this task. Depending on the discovered relationships, the features can be organized into one of the four prompt types proposed in Section 5.1. The objective would be to choose a combination of prompt types that minimizes the number of prompts in the set, thus maximizing conversational efficiency. We leave the development of such an algorithm for future research.

3. *Prompt ordering*: Depending on the prompt presentation type, the prompts may be ordered to further increase conversation efficiency. For example, the prompts may be ordered based on retrieval utility (i.e., discriminatory power, which could be estimated using information gain). Suitable methodologies for prompt ordering need to be developed.

6 Conclusion

A large number of TCBR applications need to support mixed-initiative problem solving, and conversation can help accomplish this objective. Although several research efforts have focused on conducting conversation with manually engineered cases (Aha *et al*., 2005), none has focused on conversation with automatically generated features. In this paper, we identified, formulated, and introduced the problem of conducting conversation with automatically generated features. We used a human-factors perspective of conversation in TCBR and identified four underlying issues for consideration in designing algorithms for TCBR conversation. Although previous research has measured conversation performance, it has done so only as part of system evaluation within a limited framework (e.g., Gupta & Aha, 2003; Gu & Aamodt, 2006). In this paper, we presented a model of conversation and presented measures of performance for effective and efficient conversation. Using air investigation reports as an application domain, we described a novel framework for conversation with a bag-of-words case representation.

Several methodologies require further research to address the issues raised in this paper. Namely, new methods for feature selection that meet the competing demands of case retrieval and conversation are needed, as are new methods for identifying feature interrelationships and prompt ordering. We will address these in our future research.

Acknowledgements

This research was supported by the Naval Research Laboratory.

References

Aha D.W. (1998). Textual reasoning in the context of conversational case-based reasoning systems. In M. Lenz & K. Ashley (Eds.), *Textual Case-Based Reasoning: Papers from AAAI Workshop* (WS-98-12). Menlo Park, CA: AAAI Press.

Aha, D.W., Breslow, L.A., & Munoz-Avila, H. (2001). Conversational case-based reasoning. *Applied Intelligence,* **14**(1), 9-32.

Aha, D.W., McSherry, D., & Yang, Q. (2005). Advances in conversational case-based reasoning. *Knowledge Engineering Review,* **20**(3), 247-254

Anderson, J.R. (1990). *Cognitive psychology and its implications* (3rd ed.). New York, NY: W. H. Freeman and Company.

Brüninghaus, S., & Ashley, K.D. (2005). Reasoning with textual cases. *Proceedings of the Fourth International Conference on Case-Based Reasoning* (pp 137-151). Chicago, IL: Springer.

Deerwester, S.C., Dumais, S.T., Furnas, G.W., Harshman, R.A., Landauer, T.K., Lochbaum, K.E., & Streeter, L.A. (1989). Computer information retrieval using latent semantic structure. United States Patent # 4,839,853.

Gu, M. & Aamodt, A. (2005). A knowledge-intensive method for conversational CBR. *Proceedings of the Fourth International Conference on Case-Based Reasoning* (pp. 296-311). Chicago, IL: Springer.

Gu, M. & Aamodt, A. (2006). Evaluating CBR systems using different data sources: A case study. *Proceedings of the Eighth European Conference on Case-Based Reasoning* (pp. 121-135). Fethiye, Turkey: Springer.

Gupta, K.M., & Aha, D.W. (2003). Incremental query formulation in mixed-initiative case-based reasoning. In L .McGinity (Ed.) *Proceedings of the ICCBR-03 Workshops* (pp. 172-180). Trondheim, Norway: NTNU, Department of Computer and Information Science.

Gupta, K.M., Aha, D.W., & Moore, P. (2004). Learning feature taxonomies for case indexing. *Proceedings of the Seventh European Conference on Case-Based Reasoning* (pp. 211-226). Madrid, Spain: Springer.

Gupta, K.M., Aha, D.W., & Moore P.G. (2006). Rough set feature selection algorithms for textual case-based classification. *Proceedings of the Eighth European Conference on Case-Based Reasoning* (pp. 166-181). Fethiye, Turkey: Springer.

Gupta, K.M, Aha, D.W, & Sandhu, N. (2002). Exploiting taxonomic and causal relations in conversational case retrieval. *Proceedings of the Sixth European Conference on Case-Based Reasoning* (pp. 133-148). Aberdeen, Scotland: Springer.

Massie, S., Wiratunga, N., & Craw, S. (2007). From anomaly reports to cases. To appear in *Proceedings of the Seventh International Conference of Case Based Reasoning*. Belfast, Northern Ireland: Springer.

TSB (2007). Air Investigation Reports - 2005, Retrieved from http://www.tsb.gc.ca/en/reports/ on 8 May, 2007, Quebec, Canada: Transportation Safety Board of Canada.

Wiratunga, N., Koychev, I., & Massie, S. (2004). Feature selection and generalization for retrieval of textual cases. *Proceedings of the Seventh European Conference on Case-Based Reasoning* (pp. 806-820). Madrid, Spain: Springer.

Yang, Y., & Pederson, J. (1997). A comparative study of feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412-420). Nashville, TN: Morgan Kaufmann.