Closing the Vocabulary Gap: Exploiting Unstructured Information for Intelligent Online Problem Solving

Catherine Baudin

Abstract

Companies gather valuable problem-solving knowledge in free or partially structured text that is difficult to retrieve and impossible to analyze. For example, customer complaint resolution history is often trapped as free-text notes in trouble ticket systems and cannot be leveraged to solve similar problems. The text in these repositories uses an inconsistent vocabulary that prevents both accurate retrieval and the use of otherwise powerful data mining and statistical methods. To reuse unstructured information in intelligent electronic advisory systems, we must map the text to a controlled vocabulary defined by a knowledgeable human. In this context, we can apply text-mining methods such as heuristic matching, text extraction or concept discovery techniques from raw text, in support of the data modeling process rather than as a means to directly access information. The advantages of this type of model-driven text mapping include accurate and consistent information retrieval, effective user interaction, and powerful data analysis.

Companies everywhere produce volumes of textual data in free-text documents, as notes in database tables, or in spreadsheets. Extracting useful information from these unstructured data is a challenge that only the strong of heart have successfully undertaken to date, in spite of a number of tools that have appeared in the marketplace to assist or automate this process. We will examine a key component for extracting value from unstructured information: the mapping of text elements to a normalized vocabulary.

A good example of unstructured data whose value to the enterprise is highly impaired, is the documentation of "problem / solution" history in customer service management systems. Customer support agents in call centers and in field service organizations collect descriptions of problems, symptoms and successful resolutions. In these repositories administrative information such as case numbers, dates or names of technicians are usually well formatted, but crucial problem solving knowledge is trapped in informal notes that are difficult to reuse. The potential value of these data, however, is twofold: customers and technicians could access these repositories to find solutions to problems. Companies should also be able to analyze these data to uncover trends that can be fed back to correct a product flaw, learn user preferences, or to anticipate situations.

This article discusses methods for structuring text into records that can be accurately accessed and analyzed . We use examples from customer support applications, where text is usually about problems experienced with a product line and solutions that technicians have provided. The same principles for reusing unstructured data, however, also apply to other areas ranging from the analysis of customer feedback to the exploitation of Frequently Asked Questions repositories.

Accurate Information Retrieval

Free text is notorious for its inconsistent vocabulary. As an example, a problem description for an LCD projector may refer to "a yellow spot at the center of the screen",

“a yellowish circle on the screen” or a “brown splotch right in the middle of the image” – all of which may be describing the same problem. In addition, information trapped in notes is usually riddled with abbreviations and spelling errors which adds to the difficulty. These discrepancies prevent this information from being analyzed in any robust and meaningful way and make it difficult to accurately retrieve answers from a repository of possible solutions.

Traditionally, full-text search systems have been the preferred tools for exploiting unstructured knowledge. These tools index large collections of text automatically and retrieve information by matching user requests with words from the text. Most full-text search tools now process natural language queries and accept synonyms. They are, however, more adequate for accessing large volumes of text on random subjects such as on the web, than suited to provide accurate answers to specific questions. In addition, where most intelligent human agents interact with users to clarify queries, full text search systems just produce results, good or bad, in response to the original request. As we will see, interacting with the user to clarify queries is much easier in systems based on a normalized vocabulary.

Automated Text Mining and Concept Discovery Tools

Along with traditional search, a range of text mining tools and techniques aim at analyzing collections of textual records and organize them in ways that provide insight in their content. For example, *text clustering* automatically groups documents by similarity and produce document clusters sometimes described by the sequences of words that best represent each group. The clusters, however, totally depend on word frequencies within a collection and do not take into account any explicit criteria or human input - thus providing no guarantees on the way a text collection will be partitioned.

Another class of text mining tools that operate without human supervision is based on *concept discovery*. These tools analyze text collections to extract sequences of words and relationships. These are often phrases that are repeated and are identified based on lexical and statistical criteria. Concept discovery tools can also look for proximity relations among concepts, such as groups of words that often occur together in a paragraph or a sentence. This technology has been mostly used in data visualization to let users browse and access information. It has had limited applications partly because it has been primarily geared toward open exploration rather than toward addressing specific information concerns.

These tools may uncover interesting vocabulary and relations, but *they cannot be directly used* to draw robust conclusions on relationships and trends in the data. That the phrase “yellow spot” occurs significantly more often in text referring to product P1 than to product P2 does not mean necessarily that there are more problems with spots on screens for P1. In fact it could just be that the technician who logged the information for product P1 tend to use this phrase whereas another might use “brown splotch” and “yellowish

circles" when discussing P2. In other words, there is no guarantee that trends in raw text reflect anything else than vocabulary preferences.

Automated text mining techniques that operate directly on raw text can give some idea of what a collection is about, but they do not provide accurate retrieval or robust analytical power. This type of activity is still the kingdom of information systems that are based on structured data and on *controlled vocabularies* where one concept is represented by one symbol rather than on random words with syntactic variations and synonyms. For this reason an effective way to exploit unstructured data for applications that require precise answers to questions *is to map textual information to a data model* identified by humans. Figure 1 shows part of a taxonomy of mechanical problems in cars. Concepts in this data model are short phrases that describe symptoms experienced by drivers, and relationships indicate problems that are related.

Model-Driven Text Analysis: Tools and methods for mapping unstructured text to a domain model

In model-driven text mapping input text is first matched to pre-defined phrases organized in a data model before being accessed and analyzed. This process occurs in three phases: the first phase is a model identification step where domain experts identify the data model. The second phase is the definition of the mapping methods and the addition of synonyms to bridge the gap between the text and the model. The third step is the text-processing phase where unstructured documents are associated with attributes and values form the data model. When this process occurs in batch where several thousand records are processed at once, mechanisms to efficiently detect and validate "hard cases" are essential and might constitute a fourth step.

Data Modeling and Taxonomy identification

The data model specifies what type of information is expected as input. To expand on the example in figure 1, a model of mechanical problems experienced by car drivers might contain a hierarchy of problem types for a class of vehicles along with values that refer to environmental conditions (rough road, up hill, etc.), frequencies of occurrence, etc. These models are built from different sources, by domain experts, or from existing service or training manuals. Numerical values and strings with syntactic regularities such as product or error codes on the other hand, can often be extracted directly from the input text with lexical rules and grammars (i.e. all strings starting with a p and followed by five or six digits are product codes – any word following the string *error* or *problem code* is an error code, etc.). Once these values are extracted they can be organized in the data model in hierarchies or in any other appropriate manner.

A data model usually contains attributes such as, symptoms, actions, frequency of the problem along with values and relationships. In Figure 1 "noise" is linked to "chatter", "rattle"," metallic noise," etc., through a taxonomic relation.

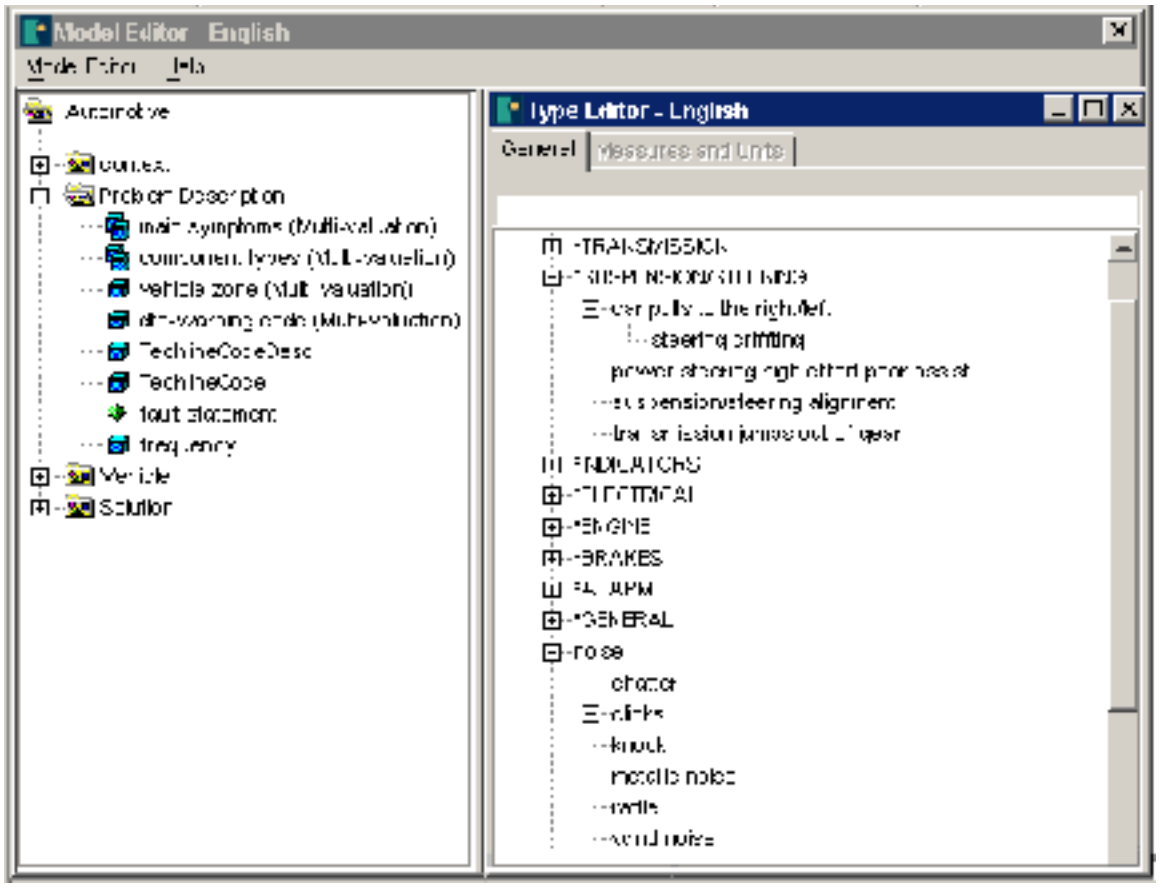


Figure 1 Taxonomy Editor

Model- driven text mapping

To bridge the gap between text and the vocabulary in the model, the text-mapping engine usually needs two main components: synonyms and *mapping procedures*. Some synonyms can be found in general thesauruses, but each domain has its own vocabulary. Spot and circle are synonyms in projector troubleshooting, but not in everyday language.

While synonyms must often be provided for each applications area or industry, many syntactic differences between a text and a model can be bridged by loosening comparison criteria for text strings. In particular, heuristic matching methods can specify what syntactic variations should be tolerated for what types of entities – people for examples can be matched primarily on last names and the associated matching procedure might recognize: John Doo, Doo J. and J. Doo as the same person. Likewise, a matching procedure may specify if partial value matches should be tolerated: can *marine blue* in the model match *blue* in the query even if the word *marine* is absent? Can “marine blue” match “marine deep blue”? Are the orders in the phrases significant? Mapping procedures can also specify the level of tolerance of the comparison to misspellings in the input. Again this level usually depends on the type of data being recognized. For example, error codes or product lines tend to be precise – chances are that E8456 is different from E8457

whereas the value “intermittently” and “intermitently” are the same even though these two strings also differ by only one character.

In general, there are many possible ways to match text to a controlled vocabulary in a model – ranging from broad *categorization* techniques to *heuristic matching*. The key to text mapping is the flexibility to associate different matching methods to different parts of a data model, so as to reflect the differences inherent to each type of data.

Finally a data model facilitates the text analysis itself because each time a value from the data model is recognized in the input, the selection narrows the set of remaining values to be mapped. Realizing that a customer problem is with *tires* automatically rules out problems with *gearboxes* and *engines* for the rest of the text mapping process. In the same way, if the model is “*Honda Accord”* then the body style cannot be *Minivan*.

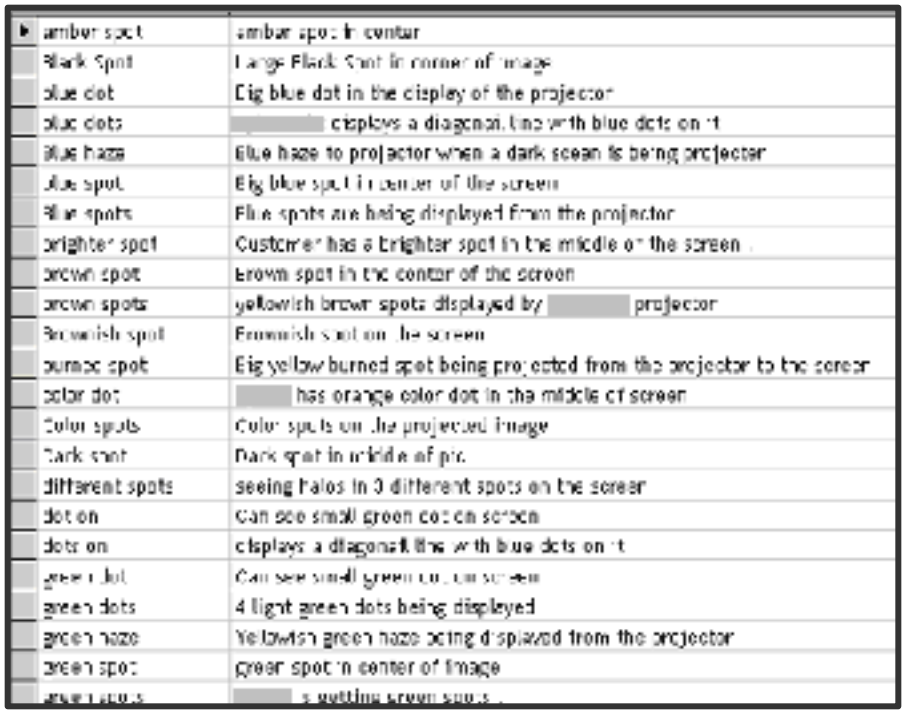
Domain-specific, model-driven text mapping is a powerful way of interpreting unstructured data. It can be done in two modes:

- Batch, where thousands of records are processed in one go.
- Interactive, where the data capture is an ongoing task.

When text mapping is performed in bulk, the data-mapping engine must provide ways to validate the results. Most text mapping engines associate a flag to “risky matches” to indicate that this result should be checked. Targeted risk assessment dramatically cuts the number of records that have to be validated. It is not possible to get 100% accuracy automatically when interpreting free-text, but the ability of a text mapping system to “get outside help” by flagging dubious results is the next best thing.

Concept discovery in support of data modeling

Automated knowledge discovery methods that operate from raw text may not be effective for direct data access or analysis, but they are promising in support of data modeling and mapping. In particular, concept discovery can highlight the wordings used in a collection in relation with values in the data model, and provide clues to vocabulary gaps, misspellings, and sometimes synonyms. Having defined the concept of “spots” with synonyms such as “dots and haze”, a concept extraction engine can highlight other words that are most often found in the vicinity of “spot” in a text collection. Figure 2 shows the result of a concept extraction where the left column shows terms related to the concept “spot on the screen” and the right column is an example of the sentence. In this case the process highlights different colors such as, amber spot, black spot, and brown and burned spot, blue haze etc. These terms can then be added to the data model as values or synonyms for possible types of spot problems for example. While concept discovery tools have been widely used in data visualization, the use of these techniques in conjunction with data modeling is still in its infancy. It is, however, a promising way to bridge the vocabulary gap in textual records when these tools are integrated with data modeling and text mapping interfaces.



amber spot	amber spot in center
Black Spot	Large Black Spot in corner of image
blue dot	Big blue dot in the cisplay of the projector
blue dots	[redacted] displays a diagonal line with blue dots on it
blue haze	Blue haze to projector when a dark screen is being projector
blue spot	Big blue spot in center of the screen
Blue spots	Blue spots are being displayed from the projector
brighter spot	Customer has a brighter spot in the middle of the screen
brown spot	Brown spot in the center of the screen
brown spots	yellowish brown spots displayed by [redacted] projector
Brownish spot	Brownish spot on the screen
burned spot	Big yellow burned spot being projected from the projector to the screen
color dot	[redacted] has orange color dot in the middle of screen
Color spots	Color spots on the projected image
Dark spot	Dark spot in middle of pic
different spots	seeing halos in 3 different spots on the screen
dot on	Can see small green dot on screen
dots on	displays a diagonal line with blue dots on it
green dot	Can see small green dot on screen
green dots	4 light green dots being displayed
green haze	Yellowish green haze being displayed from the projector
green spot	green spot in center of image
green spots	[redacted] getting green spots

Pain and rewards of model-driven text mapping

Interpretation and analysis of free-text by computers is a field where strong opinions sometimes outweigh experience and pragmatism. Data owners often believe understanding text is too hard and give up their efforts, which results in the loss of invaluable information. At the same time, methods that are based on controlled vocabularies are sometimes tainted by the perception that agreeing on a set of terms to describe a product or a situation must be too strenuous a task. In fact, the effort involved in coming up with a data model in well-defined subject areas is usually directly related to the availability and the cooperation of the people familiar with the subject. In areas that are reasonably narrow such as the support of a line of products, a usable model can usually be put together in a matter of days – because most product problems are already known. At the same time, efforts to standardize vocabulary for an industry are already underway in multiple organizations.

The advantages of bridging text with a controlled vocabulary structure results in automated information systems that combine the benefits and flexibility of free-text with the power of model-driven information retrieval. In summary, organizations can expect the following rewards:

- *Response consistency*: answers do not depend on the choice of words for describing a problem or for submitting a question. Same queries lead to same answers.
- *Accuracy and powerful similarity-based retrieval*: relations between concepts in the model are essential to uncovering similarities between queries and the data repositories. If a customer has a problem with “the engine *rattles* when the vehicle *starts*”, an

automated online assistance agent based on a data model can find that this is similar to the previously solved problem "vehicle is making strong *noises* when the *engine is cold*".

- *User interaction*: Another important aspect of model-driven user assistance is the ability of the retrieval engine to detect when a query is unclear and to interact with a user to clarify it. Because *a data model sets expectations*, the question/answer system can detect missing information. The query "my vehicle is noisy", may trigger a clarification question because the system knows that *noise* is a value in the model hierarchy that can be further specified and it can display a question such as: "Can you tell what type of noise? Rattle, Whistle, Moan…. In the same way, the system might know that the attribute "location of the problem" could not be identified in the query and might trigger a clarification question: "Where is the problem located?" "rear of the vehicle, front of the vehicle…".
- *Access to powerful analytical methods*: Finally, the data that feed the system is now associated with structured normalized values that can be analyzed with statistical and data mining techniques to further explore the data. What is the frequency of a given problem? What was the most effective way to solve it? This type of information can be used to anticipate the need for spare parts, or uncover design problems.

Conclusion

Model-driven text mapping is not a technique that applies to open-ended information retrieval systems. It shines in narrowly focused domains where it can be combined with retrieval systems based on structured data to enhance accuracy, user interaction, and data analysis, and where human interpretation can assist with modeling. Text interpretation driven by a domain model becomes easier once the model has been created and does not require a full linguistic parsing of sentences to identify key content and probable meaning in a user query. In this context, automated text mining and concept discovery tools promise to play an important role in support of data modeling and text mapping as information collection and data input channels flourish in the years to come.