Text Analytics with IBM SPSS Modeler

Table of Contents

Chapter 1 Introducing Text Analytics

Text analytics, also known as text mining, refers to the process of extracting and classifying meaningful information from unstructured text. The sources of the text itself could take many forms, from responses to open-ended questions in a survey, social media posts or emails, to sources such as engineering reports, product reviews or in-depth interviews.

The evolution of text analytics is characterized by an increasing degree of developmental sophistication in areas such as statistics, linguistics and machine learning. This is particularly true in respect to the development of approaches such as Natural Language Processing. Natural Language Processing (or NLP) is a field of research originating in the 1950's that utilised learnings from computer science, linguistics and AI with the aim of understanding passages of text in order to extract insights and information. NLP systems are designed to extract words and phrases with a view to identifying their meaning not just as adjectives or nouns but as entities such as organisations, individual's names and geographical regions.

The development of new text mining technologies in recent decades has closely followed the almost exponential growth in the digitisation of text data. So much so, that today there are literally hundreds of commercially available software programs devoted to the application of text analytics.

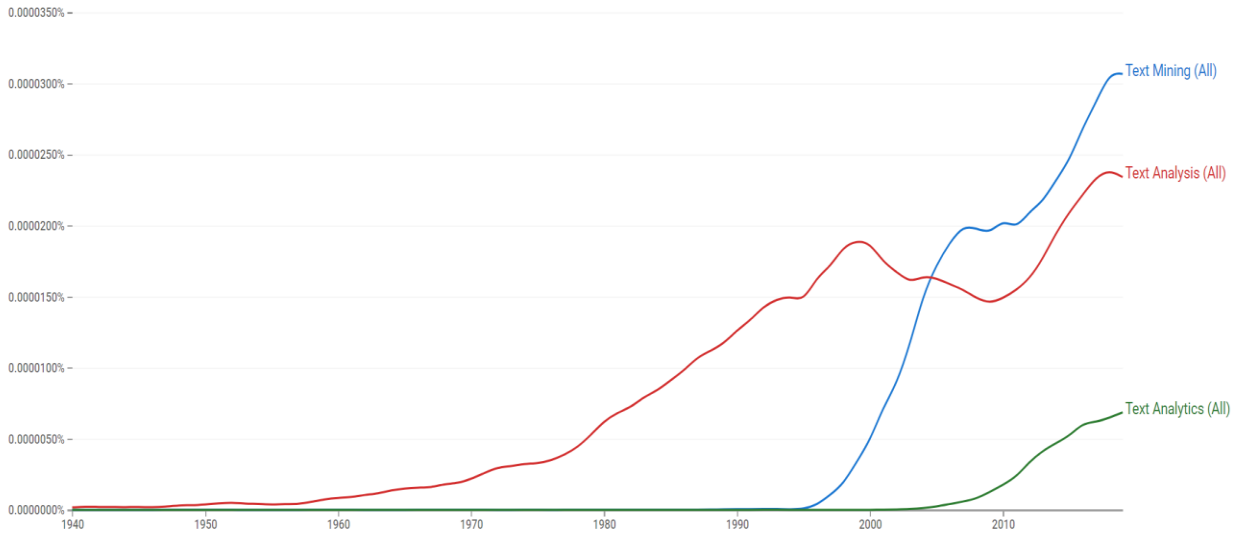


Figure 1.1 Google Ngram chart showing a marked upward trend in terms related to text mining appearing in publications since 1940

Currently, text mining applications play a crucial role in enhancing the operational efficiency and analytical prowess of thousands of organisations across the globe.

These technologies are applied in an increasingly diverse range of application areas across multiple industries such as:

- Monitoring social media for mentions of organisations, products, competitors, disease symptoms, political events and media content.
- The categorisation of documents, research papers and reports to enhance knowledge management applications.
- The analysis of consumer sentiments, voter aspirations and investor opinions, as well as clandestine activities such as potentially criminal or fraudulent behaviour.
- The creation of sophisticated predictive models that estimate the risk of asset failures, subscriber cancellation risk or the diagnosis of critical illnesses.

Despite the fact that text analytics technology is considerably more sophisticated and ubiquitous than 20 or 30 years ago, it's important to understand that ultimately these applications are focussed on making sense of language, and for even the most advanced AI systems, extracting meaning from language is hard. A language such as English contains over 170,000 words with most adult English speakers able to identify between 20,000 and 30,000 words. Moreover, the context in which a word is used makes a huge difference to the meaning of a sentence. Words such as 'break' 'cut', or 'play' have multiple definitions. Even entire phrases such as "he made her duck" have more than one interpretation. This means that extracting terms and assigning the correct contextual meaning to them is an extremely difficult exercise for computer programs to perform accurately. Indeed, this is a task that humans find hard as well. It is unlikely that two people given the job of categorising the responses from a single open-ended question in a survey of 100 people, will do so with complete agreement or in a completely consistent manner. However, this is something that text mining software program does well: so much so, that when it makes errors, it at least makes them in a very predictable manner.

The upshot of this, is that most analysts working with text analytics software should bear the following in mind:

- Mistakes will occur. The application will not be 100% accurate.
- The task of mining text is extremely iterative. Most analysts will repeatedly re-analyse the text in order to make incremental improvements.
- It's not always clear when enough is 'enough'.
- It really helps if you have a particular goal in mind. Text analytics doesn't always yield deep insights, but that doesn't mean the results are not useful.

Chapter 2 Text Analytics with IBM SPSS Modeler

Text Analytics within IBM® SPSS® Modeler is available to users as an optional add-on module, although it is currently included in the premium edition of the software as standard. It offers a host of capabilities that complement the existing statistical and machine learning functionality in the platform. This means that analysts can enhance their normal modelling activities by including fields generated as a result of mining and classifying any raw text data that might be pertinent to the application. For example, a user building a model with the aim of predicting which subscribers are likely to cancel their subscriptions to a service, can exploit not just the available quantitative measures such as age, gender, and usage but also free text information from complaints, reviews or queries.

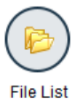
Text analytics allows analysts to turn qualitative data into quantitative information by creating a series of categories indicating, for example, if the customer had mentioned problems with billing, connection issues or simply stated how much they enjoyed the service. Alternatively, predictive maintenance applications can be developed that utilise the text from engineers' reports in order to classify the nature of faults in assets such as aircraft engines or cellular phone masts. Often text analytics is applied simply to categorise the text itself so that manufacturers can monitor trends in social media related to their products or so that crime analysts can spot common patterns in fraudulent communications.

2.1 The Text Analytics Nodes in IBM SPSS Modeler

In IBM SPSS Modeler, the text analytics functionality is accessed via a group of nodes stored in the Text Analytics tab of the node palette.

Figure 1.2 Text analytics nodes in IBM SPSS Modeler

The functionality of the various nodes in the tab can be summarised as follows:

The **File List** node 

This node is useful when working with text data that are stored in external documents rather than in a database or a structured file type. The node can scan a directory and several subfolders containing various document types such as pdfs,

Excel files, web pages and PowerPoint decks with a view to consolidating them for text analysis. To do so, the procedure generates a field containing the path address to the various documents in the archive as well as an accompanying field containing each associated document's text.

The **File Viewer** node 

The File Viewer node is designed to be used in conjunction with the previous File List node. When working with the File List node, outputting the results to a Table node means that users only see the full path name of a document rather than the text within it. Suffice to say the File Viewer is a rather old node that allows users to click on a specific document path and view the contents in a browser window.

The **Web Feed** node 

The Web Feed node allows users to read text from web sources such as blogs or news feeds in RSS or HTML formats for text analysis purposes. The node parses text into a series of fields showing content such as titles, descriptions, article and author details with each web document stored in a separate row.

The **Language** node 

The Language node is a process node which reads text sources and identifies the language it is written in. When working with large data sources containing text written in more than one language, the ability to automatically identify the source language allows the user to ensure that the correct text mining language resources are used when processing the data.

The **Text Link Analysis** node 

The Text Link Analysis node extracts concepts from the text and identifies relationships between concepts based on pre-defined pattern rules. Pattern extraction can be used to discover relationships between text concepts such as positive or negative descriptions of topics. This node offers a direct method to identify and extract text patterns before adding them to an existing dataset. As we

shall see however, we can also perform text link analysis using an interactive workbench session in the Text Mining modelling node.

The **Text Mining** node 

This is the primary tool that IBM SPSS Modeler uses to perform text analysis. The node utilises NLP methods to extract key concepts from the text, generate categories, customise text libraries, define rules and perform text link analysis to uncover relationships in the data. In many ways the Text Mining node is a functionally rich application in itself. We won't spend any time in this chapter looking at it in detail as it forms the basis for all the subsequent chapters in this course.

2.2 Using the File List and File Viewer nodes

As Figure 1.3 shows, the File List node, such as the one contained in the Modeler stream **02_File_List_WHO.str**, can be used to read a repository of text documents of various file types stored in a folder containing sub-directories.

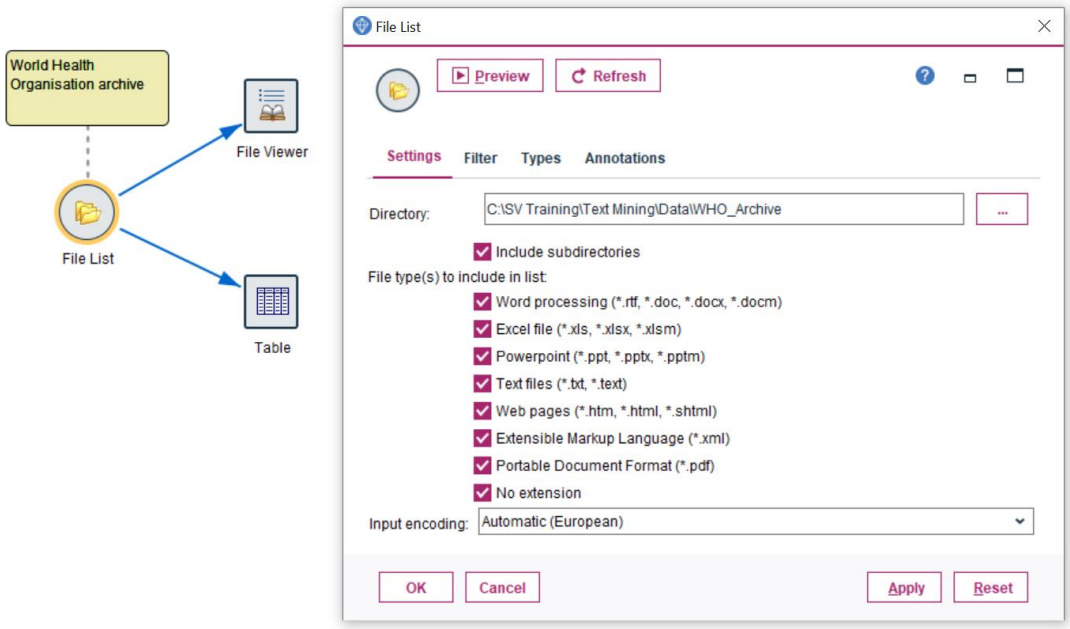


Figure 1.3 The Modeler stream '02_File_List_WHO.str' with File List and File Viewer nodes

In this example, the File List node is being used to read a number of documents collected from the World Health Organisation in November and December 2020. Figure 1.4 shows the output from this node displayed in a Table node and File Viewer

node respectively. The Table node displays both the path for an individual document in the repository as well as the text, whereas the File Viewer node shows each document path as a hyperlink that can be clicked on.

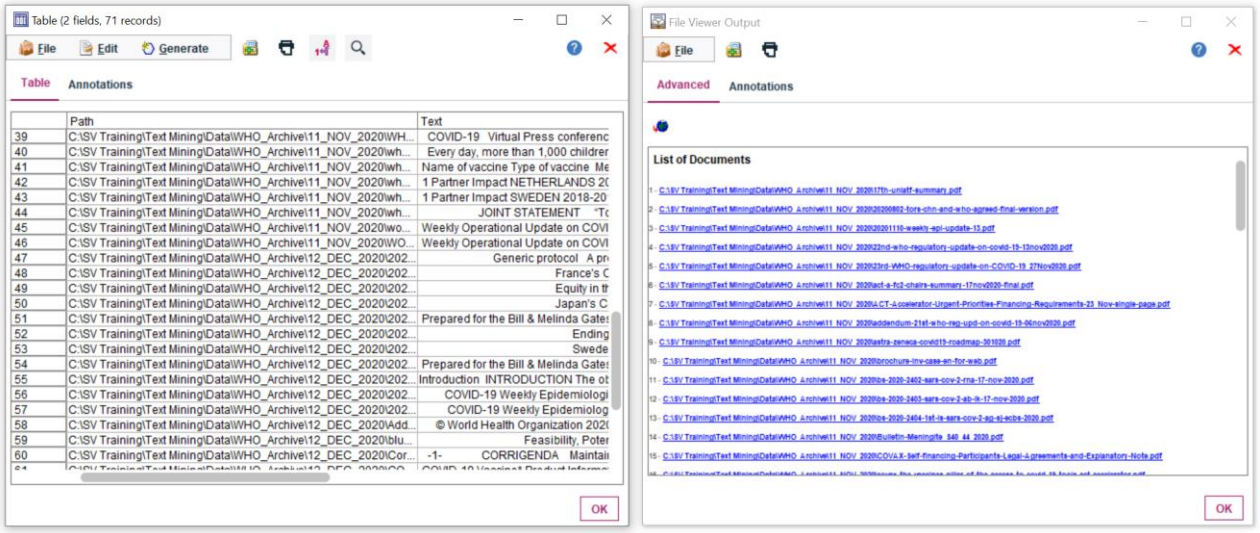


Figure 1.4 Contents of the Table node and File Viewer node showing pathways and content to documents as read by the File List node

2.3 Using the Web Feed node

Figure 1.5 shows how the Web Feed node in the Modeler stream **02_Web_Feed.str** can read text data directly from a URL address. In fact, this node is specifically designed to read data from RSS feeds. In this particular instance, the node is importing the latest news from the BBC's UK, US and International RSS feeds.

Figure 1.6 shows the Records and Content Filter tabs within the Web Feed node. The Records tab may be used to read the text content of web pages from non-RSS feeds by identifying where new records begin as well as other relevant information such as author, title and description. In reality, this tab has limited functionality and IBM suggests that to import data from *non*-RSS feeds, you may prefer using a third-party web scraping tool, such as WebQL®.

The Content Filter tab in Figure 1.6 is used to filter out unwanted information from RSS feeds. This might include copyright statements that appear in the footnotes of articles, or recurring text associated with organisational logos or slogans.

Finally, Figure 1.7 shows the resultant output from the Web Feed node displaying news information in a Table node from the three BBC RSS sites.

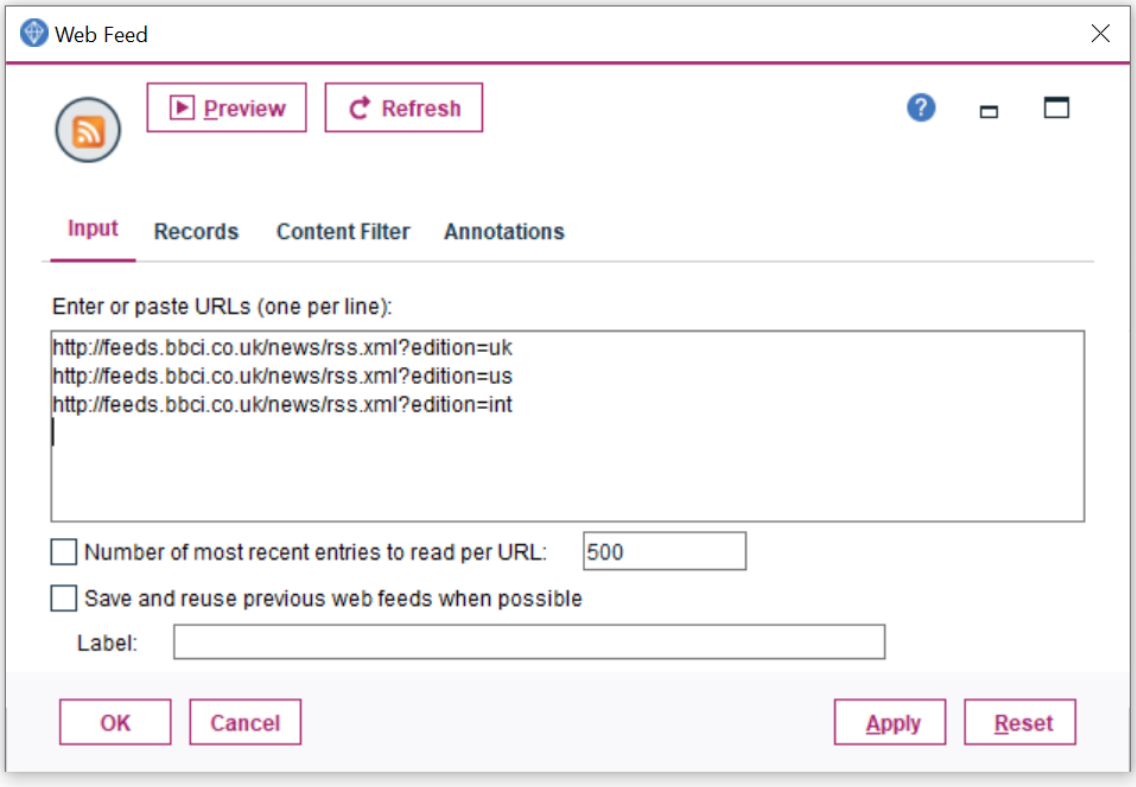


Figure 1.5 Input tab in a Web Feed node reading text data from BBC RSS new feeds

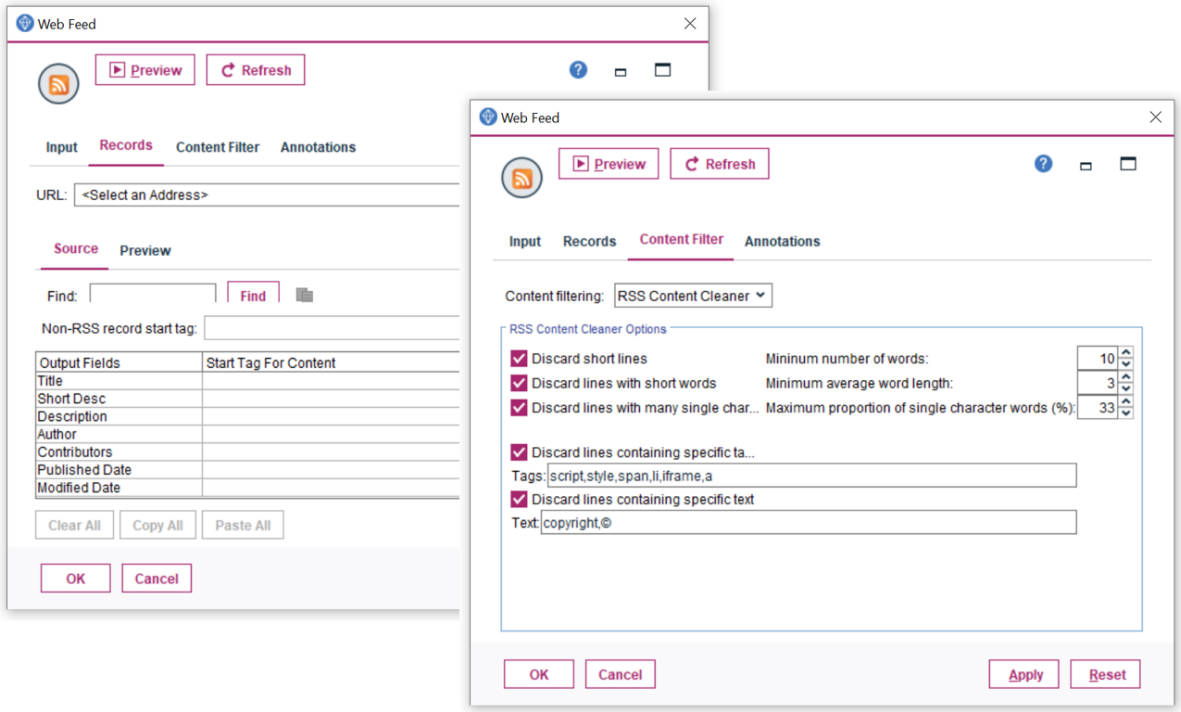


Figure 1.6 Input tab in a Web Feed node reading text data from BBC news RSS feeds

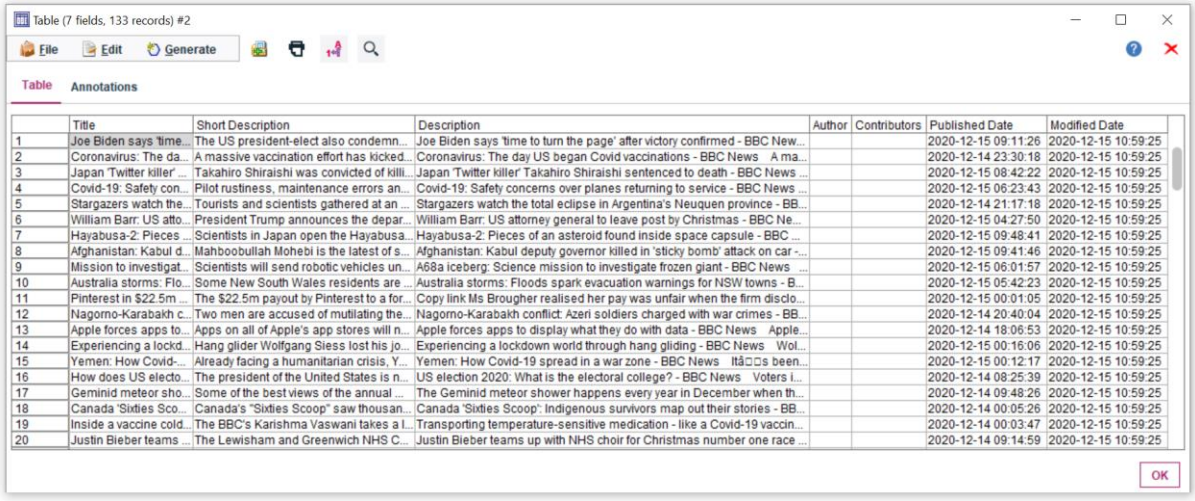


Figure 1.7 Output from the Web Feed node displaying information from BBC news RSS feeds in a Table node

2.4 Using the Language node

As Figure 1.8 shows, the Language node identifies the native language used in a body of text and creates a new field that identifies this source language using the two letter ISO language identifier code. Here we can see the Language node has correctly classified the same phrase shown in English, Italian, Portuguese, Dutch, French, Swedish and German.

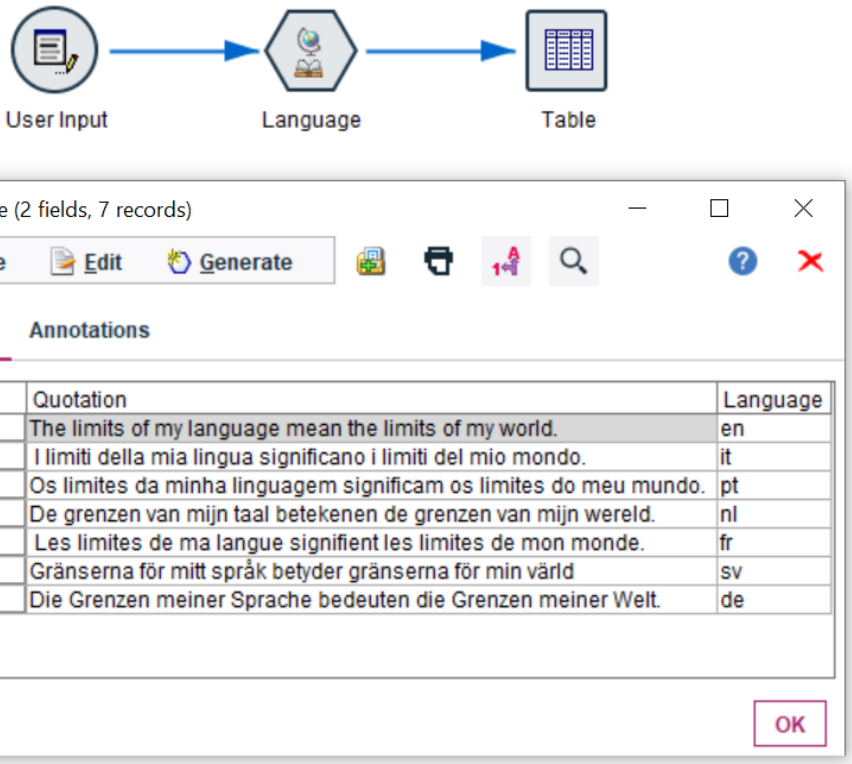


Figure 1.8 The Language node and output from the stream 02_Language_Indentifier.str

2.5 Using the Text Link Analysis node

As Figure 1.9 shows, the Text Link Analysis node is a way to identify and extract text patterns and display them in a dataset. The image displays links between such concepts as **sound** and **high-quality** as well as **design** and **cool** (look in the columns marked **Concept1** and **Concept2**). In fact, as we shall see later, existing Text Link Analysis rules can be edited, and new ones defined via an interactive workbench session in the Text Mining modelling node.

Table (15 fields, 936 records) #1

File Edit Generate

Table Annotations

	Concept1	Type1	Concept2	Type2
31	music do...	Features	not patient	NegativeAttitude
32	sound	Features	high-quality	Positive
33	sound	Features	long-lasting	PositiveFunctioning
34	sound	Features	good	Positive
35	music	Features	Null	Null
36	choice	Unknown	Null	Null
37	lack	Negative	Null	Null
38	cd player	Products	Null	Null
39	stores	Unknown	Null	Null
40	songs in p...	Unknown	compact portable	Positive
41	design	Characteristics	cool	Positive
42	software	Products	cool	Positive

OK

Figure 1.9 The Text Link Analysis node and output from the stream 02_TLA.str

Practice Exercise – Chapter 2

The default folder location for practice exercises is:

C:\SV Training\Text Mining\Student Exercises

Within the folder **Student Exercises** open the following stream:

Chapter_02_Practice.str

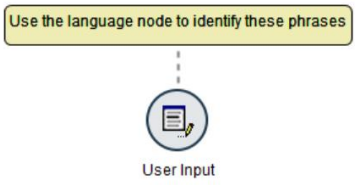
1. Attach a **Table Output** node to the **File List** node and read the contents of:

C:\SV Training\Text Mining\Data\SV_Archive

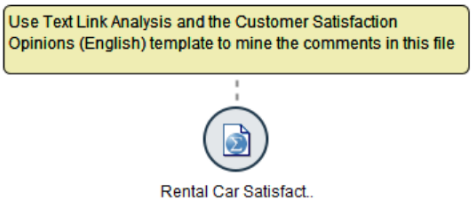
2. Attach a **Table Output** node to the **Web Feed** node and read data from:

https://news.google.com/rss

3. To the **User Input** node, attach a **Language** node *and* a **Table** node and identify the languages in the 6 example phrases.



4. To the **Statistics** node, attach a **Text Link Analysis** node *and* a **Table** node and perform a text link analysis on the source node's dataset.



Chapter 3 The Text Analysis Process

Within IBM SPSS Modeler, text mining takes the form of an end-to-end process, beginning with importing data and ending with the creation of a text mining model that can be used to read and categorise any new relevant text data as and when it becomes available. This overall process is illustrated in Figure 3.1.

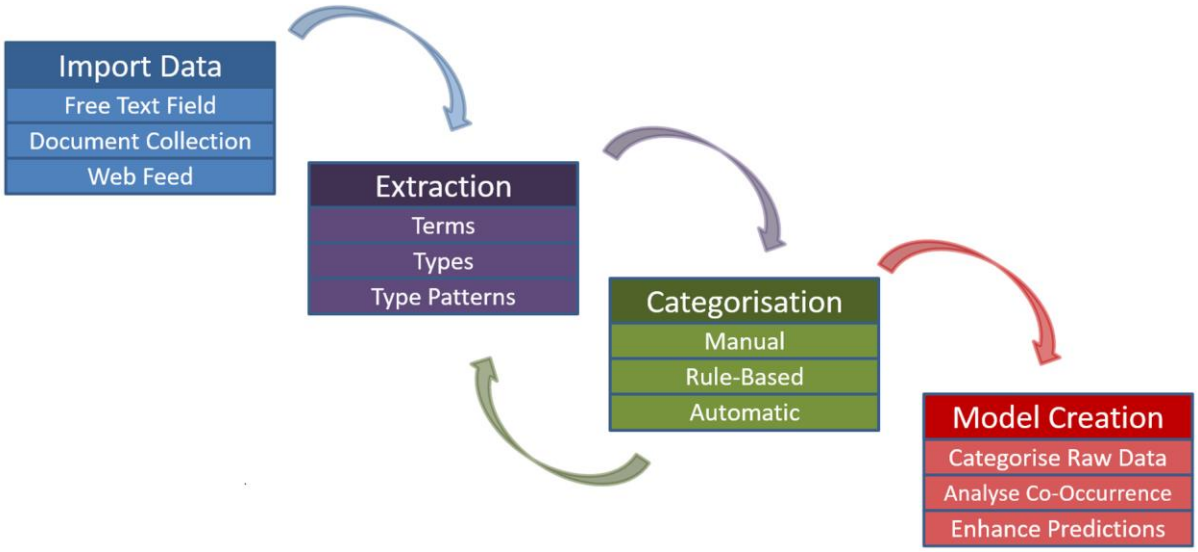


Figure 3.1 The Text Mining process in IBM SPSS Modeler

3.1 Importing Data

IBM SPSS Modeler can read a wide range of data file types. As such, text information might appear as a comment or response within a field stored in an ASCII text file, an MS Excel workbook, a database, or a JSON file. Furthermore, we've already seen that Modeler also offers the ability to read text data stored in a nested repository of various document formats or as RSS web feeds. This ability to read and load the text data is, of course, a necessary pre-requisite to any kind of analysis and so it helps that Modeler not only provides a number of ways to deal with this task, but also has an extensive array of tools to help merge, clean and transform the data prior to any analysis taking place.

3.2 Extraction

Extraction represents a critically important and multi-faceted stage of the text analytics process in Modeler Text Analytics. It's important to understand that by default, not all of the text is deemed as useful or relevant. With this in mind, the extraction stage consists of several routines that Modeler Text Analytics performs to identify and extract the most pertinent and valuable words and expressions for any further analysis or categorisation.

As we know, the correct interpretation of language is highly dependent on the context in which it is used. This is an especially important aspect of the many challenges that text mining encounters, as often the same word in a language can be used as a noun, an adjective or as a verb.

Consider the following sentences:

- At the end of the play (*noun*), Alex would play (*verb*) the piano.
- When her fast (*noun*) was over, she caught the fast (*adjective*) train home.
- He needed help to tie (*verb*) his tie (*noun*)

NLP software, like Modeler Text Analytics, is developed with a keen understanding of linguistic analysis. This involves an appreciation of the elements, structure, and meaning of language. In linguistics, **morphology** refers to the study of the smallest units of meaning for individual words or *morphemes*. The word *judge*, for example, is a morpheme of a larger collection of words such as *judges, judging* or *judgement*. Linguistic **syntax** on the other hand, is focussed on the set of rules, principles, and processes that govern the structure of sentences. Phrases such as **the shark the man ate** are highly dependent on word order to be interpretated correctly, for the simple reason that the phrase **the man the shark ate** yields a completely different meaning. Finally, **semantics** relates to studying the actual meaning of words and phrases. The expression **to break** has several meanings including to damage, to take a rest and to impart difficult information. Even entire sentences such as **we should discourage demeaning work** can have two interpretations.

3.2.1 Parts of Speech

One aspect of how Modeler Text Analytics attempts to make sense of text data, is by employing a **Parts of Speech** algorithm. Parts of speech refers to the categories to which words are assigned in accordance with their syntactic functions. In English, the main parts of speech are noun, pronoun, adjective, determiner, verb, adverb, preposition, conjunction, and interjection.

Modeler Text Analytics attempts categorise written text in a similar fashion. It does this by tagging words according to its own internal parts of speech (POS) schema. The following shows a list of parts of speech codes that the software employs during the extraction phase of the text analytics process.

N: Noun

> A word that is the name of something (such as an object, animal, place, function, quality, idea, or action).

A: Adjective

Adjectives are describing words that are related to an attribute of a noun, such as ***green***, ***happy***, ***large***, or ***mechanical***.

V: Verb

A *doing* word that describes an action, state or occurrence. Such as the act of running e.g., **Alex *ran* to the store**.

B: Adverb

An adverb is a word or an expression that describes a verb or adjective. Adverbs are typically used to express manner, frequency, intensity, or level of certainty e.g., **Robin *quickly* replied**.

P: Participle

A verb form that can be used as an adjective. Participles usually end in ***ed*** or ***ing*** and often refer to an action e.g., **She was *listening*.**

S: Stop word

In Modeler Text Analytics, **Stop** words refer to an extensive list of words that, by default, are excluded from the extraction process. Stop words include all pronouns and prepositions (except the word ***of***).

C: Preposition

A word that tells you when or where something is, in relation to something else. These are often short words such as ***from***, ***of*** and ***for***. Prepositions are usually tagged as S or Stop words.

D: Determiner

A word placed in front of a noun to specify a quantity or clarify what the noun is referring to. Words such as ***any***, ***many***, ***my*** or ***these*** are determiners e.g., **we visited *their* home.**

G: Gerund

A noun that is derived from a verb. Like a participle verb form, this is another kind of word that ends in ***ing*** but this time it is behaves as a noun e.g., **he enjoyed the *training***.

O: Coordination

> Coordinating conjunctions are words that link parts of a sentence together. These include words such as ***and***, ***or***, ***but*** and ***yet***.

X: Auxiliary

> Often referred to as ***helping*** verbs. Auxiliary verbs add functional or grammatical meaning to a clause. Words such as ***is***, ***have***, ***can***, ***could***, or ***will*** usually accompany a main verb e.g. **He *is* sleeping**.

Now that we've identified the parts of speech that Modeler Text Analytics uses to tag words in text data, let's look at how it would deal with the following sentence:

> ***Analytical technology from Smart Vision Europe Ltd can help your company to discover insights in data and develop applications to predict future conditions and find new opportunities***

Figure 3.2 shows how parts of speech tagging would parse this sentence.

Analytical	*technology*	*from*	*Smart*	*Vision*	*Europe*	*Ltd*	*can*	*help*
A	**N**	**X**	**NA**	**N**	**N**	**N**	**X**	**V**
your	*company*	*to*	*discover*	*insights*	*in*	*data*	*and*	*develop*
X	**N**	**X**	**V**	**N**	**X**	**N**	**O**	**V**
applications	*to*	*predict*	*future*	*conditions*	*and*	*find*	*new*	*opportunities*
N	**X**	**V**	**NA**	**NV**	**O**	**NV**	**A**	**N**

Figure 3.2 Example of parts of speech tagging

Notice how:

- Words such as ***future*** and ***Smart*** are tagged as adjectives *and* nouns (NA).
- The words ***conditions*** and ***find*** are tagged as nouns *and* verbs (NV).
- The words ***from***, ***can***, ***your***, ***to*** and ***in*** are tagged as Stop words

It's worth noting that when the system detects a word that can be viewed both as a noun or adjective, or as a noun and a verb, the software generally treats it as a noun.

When the system encounters a word beginning with an uppercase letter that it doesn't recognise, such as an acronym or product name, it tags it with a **?** character and treats it as a noun.

3.2.2 Extraction Patterns

The reason Modeler Text Analytics employs parts of speech tagging, is that it plays a critical role in deciding which text strings are extracted and which are ignored. To do

this, it uses parts of speech tagging in conjunction with a series of **extraction pattern rules** that control the extraction of phrases with multiple words. Based on the software's default set of extraction pattern rules, when our example sentence is submitted to Modeler Text Analytics, the following nine concepts are extracted:

1. **analytical technology**
2. **smart vision europe ltd**
3. **help**
4. **company**
5. **insights**
6. **data**
7. **applications**
8. **future conditions**
9. **new opportunities**

New users of the software soon notice that the extraction process tends to be heavily focussed on nouns. In our example sentence, the words ***discover*** and ***develop*** were not extracted as the system generally ignores verbs (although this can be overwritten). You can also see that the system has extracted the concept **analytical technology**. Extracted terms comprised of more than one word are referred to as ***compound terms***. Extraction patterns select compound terms because these phrases contain word types that match some pre-specified combination of syntax. A typical extraction pattern looks for an adjective followed by a noun such as **analytical technology** or **new opportunities**. Other patterns look for an adjective followed by a series of nouns such as **Smart Vision Europe Ltd**. Indeed, compound terms like **future conditions** may match more than one extraction pattern, as the words can be viewed as adjective and noun, noun and noun, adjective and verb or noun and verb.

Some examples of compound terms and their associated extraction pattern rules are shown in Figure 3.3.

Compound Term	Extraction Pattern
personal digital assistant	adjective – adjective – noun
liquid crystal display	noun – noun – noun
wireless local area network	adjective – adjective – noun – noun
printed invoices	past participle – noun
osteoarthritis of the hip	noun – preposition – determiner – noun

Figure 3.3 Compound terms and associated extraction pattern rules

3.2.3 Synonyms and substitution

As much of the extraction process is devoted to the identification of candidate terms, it's important to realise that the software's ability to recognise a term is dependent on its use of various language resources. It's possible to augment and edit these resources by adding new terms. In fact, Modeler Text Analytics ships with a portfolio of resource templates containing terms and phrases that are specific to certain industries or applications such as banking or customer satisfaction. When working with these resources, it's not unusual to discover that the system has extracted a term that is actually a synonym of a particular word in the text. This is because the software will swap a candidate term for a target synonym if that word happens to appear in the **substitution dictionary** of the relevant library resource. In Figure 3.4 we can see which extracted concepts are displayed when the following sentence was submitted to Modeler Text Analytics using the included **Customer Satisfaction Opinions** resource template:

The service was speedy and cordial

Note that the words ***speedy*** and ***cordial*** have been swapped for the synonyms ***fast*** and ***courteous***. This is because the replacement words appear as targets in the resource template's substitution dictionary.

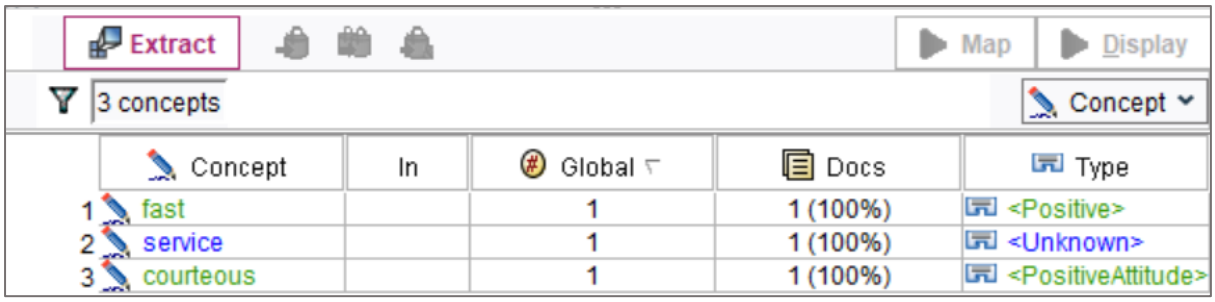


Figure 3.4 The extracted terms appearing as synonyms of words in the text due to the use of a substitution dictionary

3.2.4 Equivalence classes

Having used parts of speech tagging and extraction patterns to identify candidate terms, the extractor system applies a series of sophisticated algorithms to identify phrases that have the same meaning. Such terms are said to belong to the same **equivalence class** and identifying them as such, means that they are not extracted as separate concepts. There are a number of ways in which the algorithm identifies equivalence classes.

- **Geographical/cultural variations:** Where words such as ***labor*** and ***labour*** are treated as identical.
- **Variation in separators:** Terms such as ***user-friendly*** and ***user friendly*** are extracted as ***user-friendly***.
- **Component permutations:** ***Users of the software*** and ***software users*** are extracted as ***software users***. The rules governing the extracted term are based on:
    - User-specified synonym
    - The most frequent form of the term
    - The shortest form of the term
    - The first one that is encountered
- **Inflected terms:** Pluralised terms such as ***Software users*** are treated as equivalent to the singular expression ***software user***.
- **Fuzzy grouping for spelling mistakes:** This routine works by temporarily removing all vowels (except for the first vowel) as well as double or triple consonants from extracted words before comparing them to see if they are the same. So terms such as ***accommodation***, ***acommodation*** and ***accomodation*** would be recognised as the same word.

It's worth noting that the fuzzy grouping algorithm can result is a term being incorrectly extracted. If, for example, the system does not recognise the word ***stationary*** but does recognise the word ***stationery*** it may mistake the former term for the latter. As we shall see, there are a couple of methods that users may employ to correct these kinds of errors.

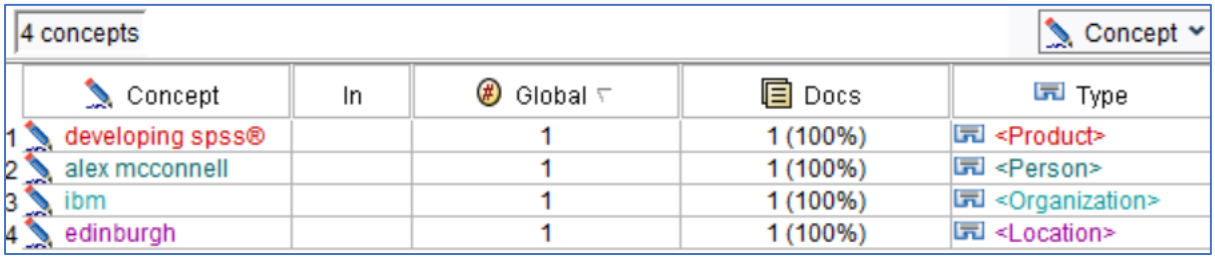
3.2.5 Assigning Types

Type assignment is one of the most important aspects of working with Modeler Text Analytics. This is because term types are so useful in making sense of the themes within the data, especially when we aim to categorise the text or discover relationships between topics. A term type is a higher-level concept that contains one or more terms. For example, the term ***apple*** might be associated with the term type ***fruit***. As we know, the software comes pre-packaged with a portfolio of resource templates that contain various libraries of terms and types. In turn, each resource template is comprised of a series of **Type Dictionaries**. The type dictionaries are collections of words and phrases grouped under various term types. For those working with English language text, the default resource template that Modeler Text Analytics employs is called **Basic Resources (English)**. This contains the core libraries for that language and as such, it forms the foundation for all the other language equivalent resources available to the user. The Basic Resources template includes type dictionaries for people, organisations, products and locations. It should be noted that the dictionaries in the Basic Resources template represent a special case

that are referred to as ***compiled* resources**. This means that although they only display a few examples of the terms within each type, in reality the software has a much larger, albeit hidden, list of terms that it can draw upon. Users are also able to add the names of people, products, locations and organisations to existing these resources if the system doesn't recognise a term.

Figure 3.5 shows the how the software assigns types to the terms extracted from the sentence:

Alex McConnell works for IBM in Edinburgh developing SPSS® software



4 concepts — Concept

	Concept	In	Global	Docs	Type
1	developing spss®		1	1 (100%)	<Product>
2	alex mcconnell		1	1 (100%)	<Person>
3	ibm		1	1 (100%)	<Organization>
4	edinburgh		1	1 (100%)	<Location>

Figure 3.5 Extracted terms assigned to the four core term types using the Basic Resources (English) template

Although the Basic Resources template may contain only four primary term types, this basic functionality can be greatly enhanced by using an additional template such as the **Opinions** template. The Opinions template provides its own library comprising an extensive list of term types devoted to capturing expressions of *sentiment* in text. Examples of these additional term types include those that identify positive terms as well as phrases related to positive attitudes or positive functioning. Conversely, the library also contains term types related to negative phrases and negative feelings.

3.2.6 Type Patterns

A special form of extraction is **Text Link Analysis** where Modeler Text Analytics applies a series of rules to extracted terms in order to establish relationships between topics that co-occur in the text. It is possible for users to edit the software's advanced resources in order to create their own text link analysis rules. Figure 3.6 shows the results of a text link analysis extraction used in conjunction with the Customer Satisfaction Opinions resource template and applied to the following sentences:

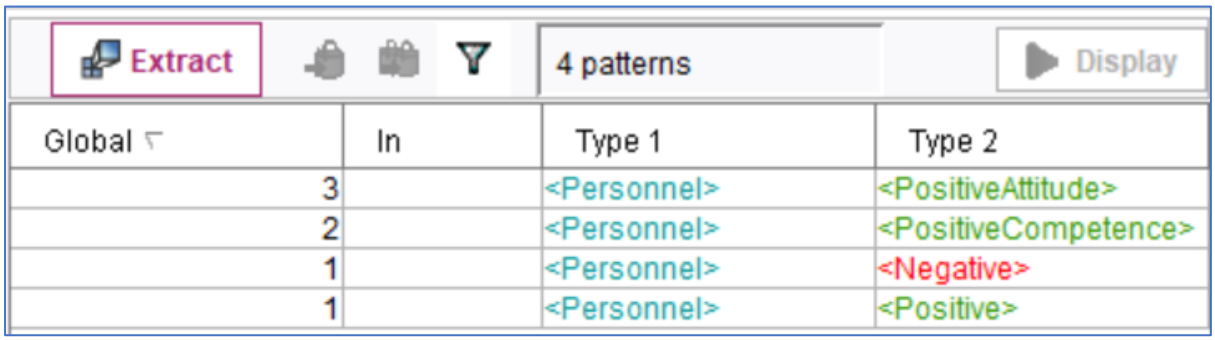
We stood for a long time at the front desk

The manager was friendly and polite

The staff really helped us

The reception staff were very professional

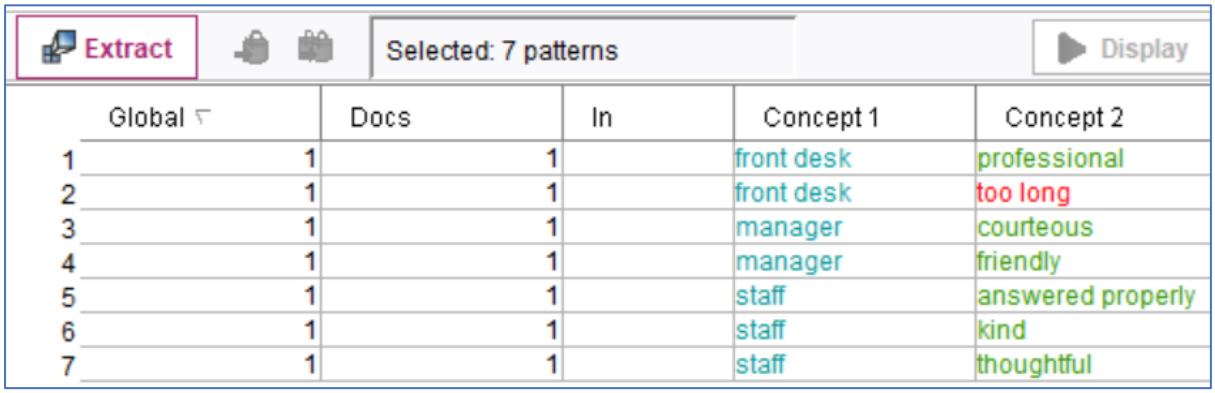
I generally found the staff to be kind and thoughtful



Global	In	Type 1	Type 2
3		<Personnel>	<PositiveAttitude>
2		<Personnel>	<PositiveCompetence>
1		<Personnel>	<Negative>
1		<Personnel>	<Positive>

Figure 3.6 Text Link Analysis showing links between term types

The links between the actual extracted terms are shown in Figure 3.7 (note that the extracted term ***helped us*** has been substituted by the concept ***answered properly*** just as the phrase ***reception staff*** has been substituted for the concept ***front desk***).



	Global	Docs	In	Concept 1	Concept 2
1	1	1		front desk	professional
2	1	1		front desk	too long
3	1	1		manager	courteous
4	1	1		manager	friendly
5	1	1		staff	answered properly
6	1	1		staff	kind
7	1	1		staff	thoughtful

Figure 3.7 Text Link Analysis showing links between extracted terms (referred to here as 'concepts')

3.3 Categorisation

Categorising responses is a process that normally occurs once the analyst is confident that the system has extracted all the relevant terms and assigned them to the appropriate term types. Later in the course, we will take a closer look at how categorisation is performed. For now, we need only to understand that categorisation refers to the creation of categories that aim to classify the nature of each text response. In a simple survey of 500 people asked about their level of satisfaction with a new phone, their responses might refer to multiple topics across a

range of sentiments. It's easy to imagine that these responses might in turn fall into a number of categories such as ***Poor battery life***, ***Attractive design*** or ***Easy to use***.

Modeler Text Analytics offers the user several methods for creating categories:

- Creating a category for a single extracted concept such as ***sound quality***
- Creating a category for a term type such as ***Speed*** containing multiple terms like ***fast***, ***quick***, ***speedy***, ***rapid*** or ***swift***
- Using a custom-built rule such as ***charging & time*** to populate a category called ***Charging speed***
- Employing Text Link Analysis to automatically discover links between terms or types and using the uncovered links to create new categories
- Forcing an individual response into an existing category
- Using an automatic categorisation method that create categories and assign cases based on one (or both) of Modeler Text Analytics' built-in algorithms

Having created some initial categories, the user assesses how many cases have been assigned to each category and decides whether to delete, merge or rename any of the existing categories. They also look at the responses that remain uncategorised in order to decide if any can be added to the existing categories or require new ones to be created. As with much of text analytics, this is a highly iterative process that may require multiple passes of the data. Figure 3.8 shows an image containing the categories from a partially categorised survey where respondents have been asked what they like *least* about a new mp3 player.

Category	Descriptors	Docs
All Documents	-	405
Uncategorized	-	134
No concepts extracted	-	3
battery	15	75
expensive	3	49
music & songs	15	36
nothing wrong	1	36
bulky / heavy	4	28
storage & memory	20	27
ear/head phones	9	15
color	5	13
small size	2	11
downloading / uploading	8	8

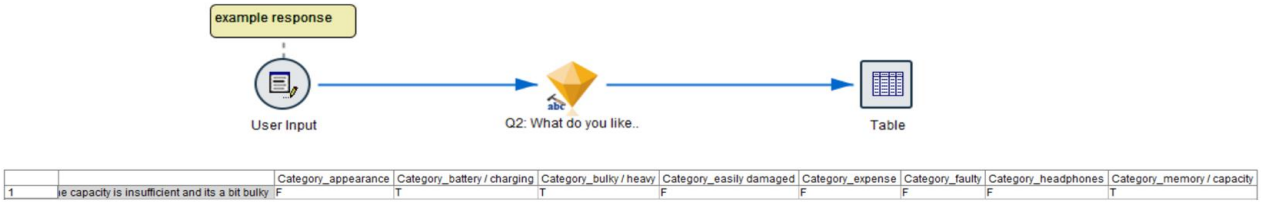
Figure 3.8 Categories created in Modeler Text Analytics to classify responses to a product survey

3.4 Model Creation

Much of IBM SPSS Modeler is focussed on the development of analytical models that support predictive applications. Modeler Text Analytics also allows the user to create models. The text mining models can then be used to classify new text data

that the model has been trained to recognise. Figure 3.9 shows how a text mining model has be used to categorise the following sentence:

The battery drains too fast. The capacity is insufficient and it's a bit bulky



		Category_appearance	Category_battery / charging	Category_bulky / heavy	Category_easily damaged	Category_expense	Category_faulty	Category_headphones	Category_memory / capacity
1	ie capacity is insufficient and its a bit bulky	F	T	T	F	F	F	F	T

Figure 3.9 Text mining model used to score and categorise an example sentence

In the example, we can see how each of the fields **Category_battery / charging**, **Category_bulky/heavy** and **Category_memory / capacity** are marked with the letter **T** indicating **true** as these are the three topics included in the sentence.

Practice Exercise – Chapter 3

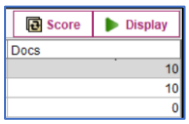
Within the folder **Student Exercises** open the following stream:

Chapter_03_Practice.str

1. Right-click on the **Table** node with the label **10 Records** and run this stream branch to view the sample data. Look at the range of terms and expressions the respondents have used.

2. Now right-click on the **Text Mining** node labelled **Example 1** and run this branch. Here the text mining node is using only the **Basic Resources** template. Look at the terms that have been extracted. Notice if any of them should belong to a type group other than **Unknown**.

Near the top of the **Categories and Concepts** window, click the row marked **All Documents** and click the **Display** button.



Look at the response data and notice any terms and phrases that were *not* included in the extraction process.

Close the and **Exit** the session without updating the node.

3. Now, right-click on the **Text Mining** node labelled **Example 2** and run this branch. Here the text mining node is using the **Opinions** resource template. Repeat the steps as above and notice the differences in the extraction results from previously.

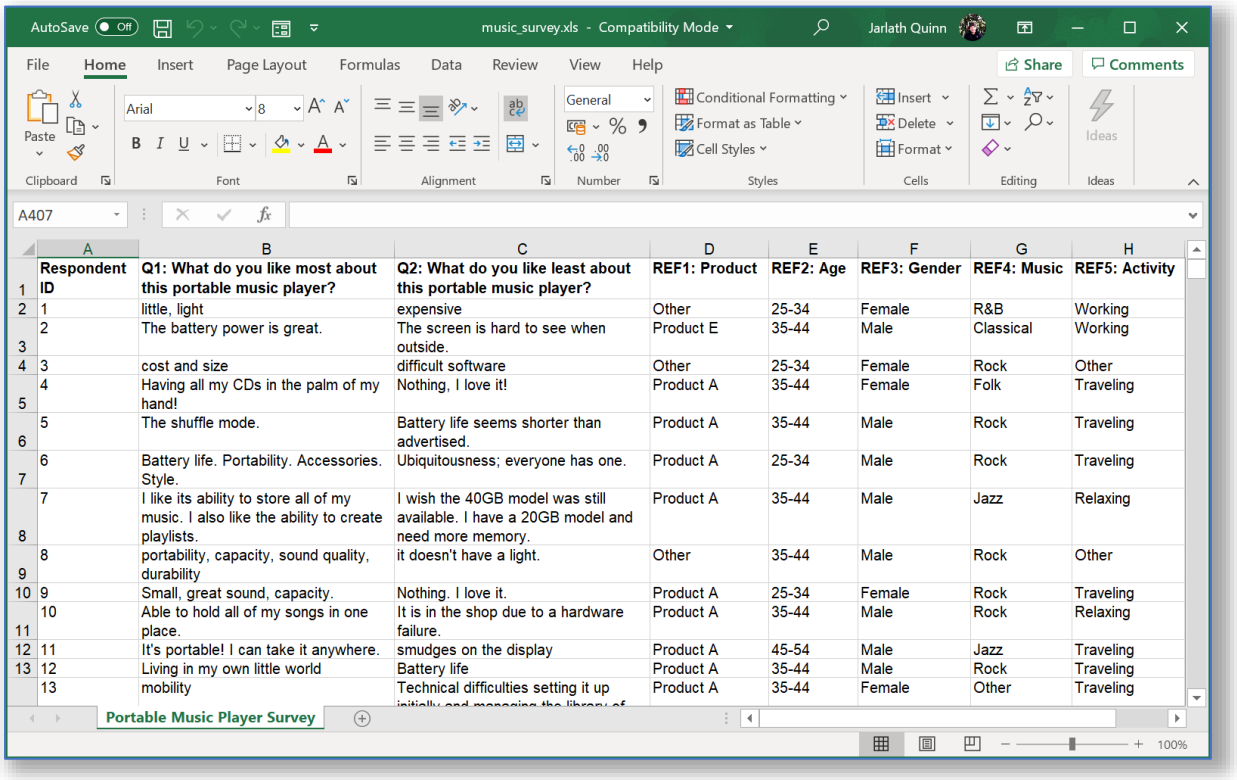
Close the and **Exit** the session without updating the node.

4. Right-click on the **Text Mining** node labelled **Example 3** and run this branch. Click **OK** on the initial warning dialog. Again, repeat the previous steps. Notice that this iteration also includes a number of pre-built text categories. **Close** the and **Exit** the session without updating the node.

Right-click on the **Table** node with the label **Example 4** and run this branch. Look at the single phrase and text categories created by the text mining model. Has the model captured the topics in the phrase? Double click on the model and browse its contents. Feel free to edit the model and experiment with the results. **Close** the and **Exit** the session without updating the node.

Chapter 4 The Text Mining Node

In this chapter we will take a closer look at the Text Mining node. Returning to the example we introduced in the previous chapter, Figure 4.1 shows the contents of the data file **music_survey.xls**. The file contains verbatim responses from 405 people asked to evaluate what they liked and disliked about a new mp3 player. To analyse this dataset using Modeler Text Analytics, we need to read the file using an Excel source node and then specify which of the two open-ended questions we wish to extract data from. Figure 4.2 shows the stream **04_The_Text_Mining_Node.str** that enables this.



Respondent ID	Q1: What do you like most about this portable music player?	Q2: What do you like least about this portable music player?	REF1: Product	REF2: Age	REF3: Gender	REF4: Music	REF5: Activity
1	little, light	expensive	Other	25-34	Female	R&B	Working
2	The battery power is great.	The screen is hard to see when outside.	Product E	35-44	Male	Classical	Working
3	cost and size	difficult software	Other	25-34	Female	Rock	Other
4	Having all my CDs in the palm of my hand!	Nothing, I love it!	Product A	35-44	Female	Folk	Traveling
5	The shuffle mode.	Battery life seems shorter than advertised.	Product A	35-44	Male	Rock	Traveling
6	Battery life. Portability. Accessories. Style.	Ubiquitousness; everyone has one.	Product A	25-34	Male	Rock	Traveling
7	I like its ability to store all of my music. I also like the ability to create playlists.	I wish the 40GB model was still available. I have a 20GB model and need more memory.	Product A	35-44	Male	Jazz	Relaxing
8	portability, capacity, sound quality, durability	it doesn't have a light.	Other	35-44	Male	Rock	Other
9	Small, great sound, capacity.	Nothing. I love it.	Product A	25-34	Female	Rock	Traveling
10	Able to hold all of my songs in one place.	It is in the shop due to a hardware failure.	Product A	35-44	Male	Rock	Relaxing
11	It's portable! I can take it anywhere.	smudges on the display	Product A	45-54	Male	Jazz	Traveling
12	Living in my own little world	Battery life	Product A	35-44	Male	Rock	Traveling
13	mobility	Technical difficulties setting it up	Product A	35-44	Female	Other	Traveling

Figure 4.1 Contents of the file music_survey.xls

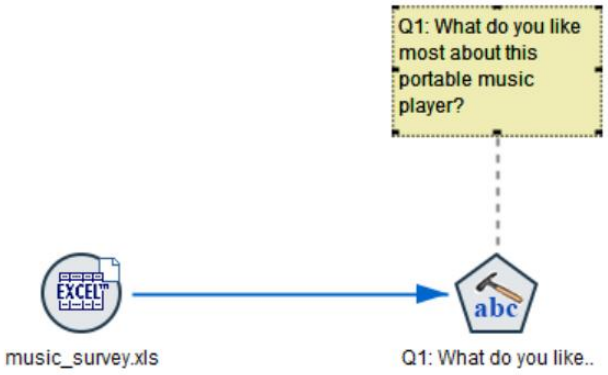


Figure 4.2 Contents of stream 04_The_Text_Mining_Node.str

4.1 Introducing the Text Mining Node

Double-clicking on the Text Mining node allows us to access the settings that control the initial extraction process. Figure 4.3 shows an image of the **Fields** tab in the node with numbered annotations.

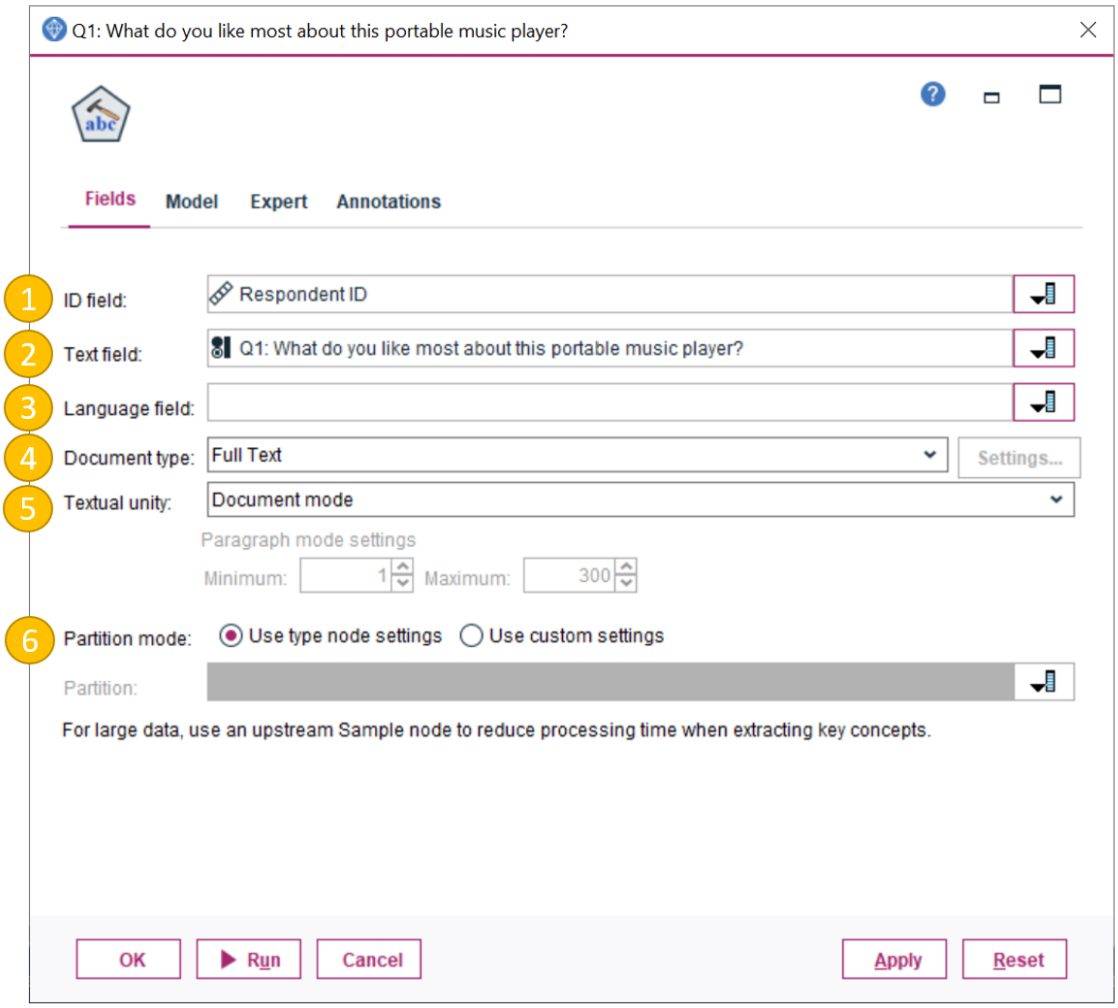


Figure 4.3 Fields tab in the Text Mining node

The currently displayed Fields tab contains the following controls:

1. **ID Field**: This is an optional setting which allows the user to select an existing integer field containing an ID number for each respondent. This option makes it easier for users to identify individual records when working within the text mining workbench.
2. **Text Field**: This drop-down menu allows the user to choose which text field the application should use as the basis of for the text mining exercise. As you might expect, this is a mandatory specification.
3. **Language Field**: As Chapter 2 showed, Modeler Text Analytics includes a Language node which can automatically detect different languages and

generate a new field identifying the source language using a two letter ISO language identifier code. It is not necessary to use this option if each response is in the same language as the currently selected resource template.

4. **Document Type:** The term *document* here indicates that this option is more relevant when using the File node to read a directory of document types.
    a. **Full Text:** This option is normally used for non-tagged documents such as Word and PowerPoint and PDF files.
    b. **Structured Text:** This option is designed for text pages that contain regular structures such as research papers with titles, authors and abstracts. Choosing this option enables the **Settings** button which allows the user to define text separators in the **Structured Text Formatting** window. It's especially useful if one wants to exclude certain sections of the text from extraction as any section not declared is ignored.
5. **Textual Unity:** Again, these controls make sense when dealing with documents rather than response fields.
    a. **Document Mode:** This mode simply treats the entire document as a single entity to be mined. It should be used for articles and reports that are relatively homogenous.
    b. **Paragraph Mode:** If this mode is selected, the extraction process is applied on a paragraph-by-paragraph basis. In other words, the procedure treats each paragraph as if it was a separate document. This is useful when dealing with reports and documents that range across many different subjects. Paragraph mode does not work with PDF documents.
6. **Partition mode:** Partitioning data is a common technique used in many model building nodes throughout IBM SPSS Modeler. Data partitions are usually indicated with a field that randomly marks the observations as **Training** or **Testing**. If the analyst has a sufficient amount of sample data, they can build a model on the Training group with a view to seeing how similar the results are, or how accurately it classifies the data in the Testing group.

Clicking on the **Model** tab provides access to a number of settings controlling the extraction process itself. Figure 4.4 shows an image of the Model control tab with numbered annotations.

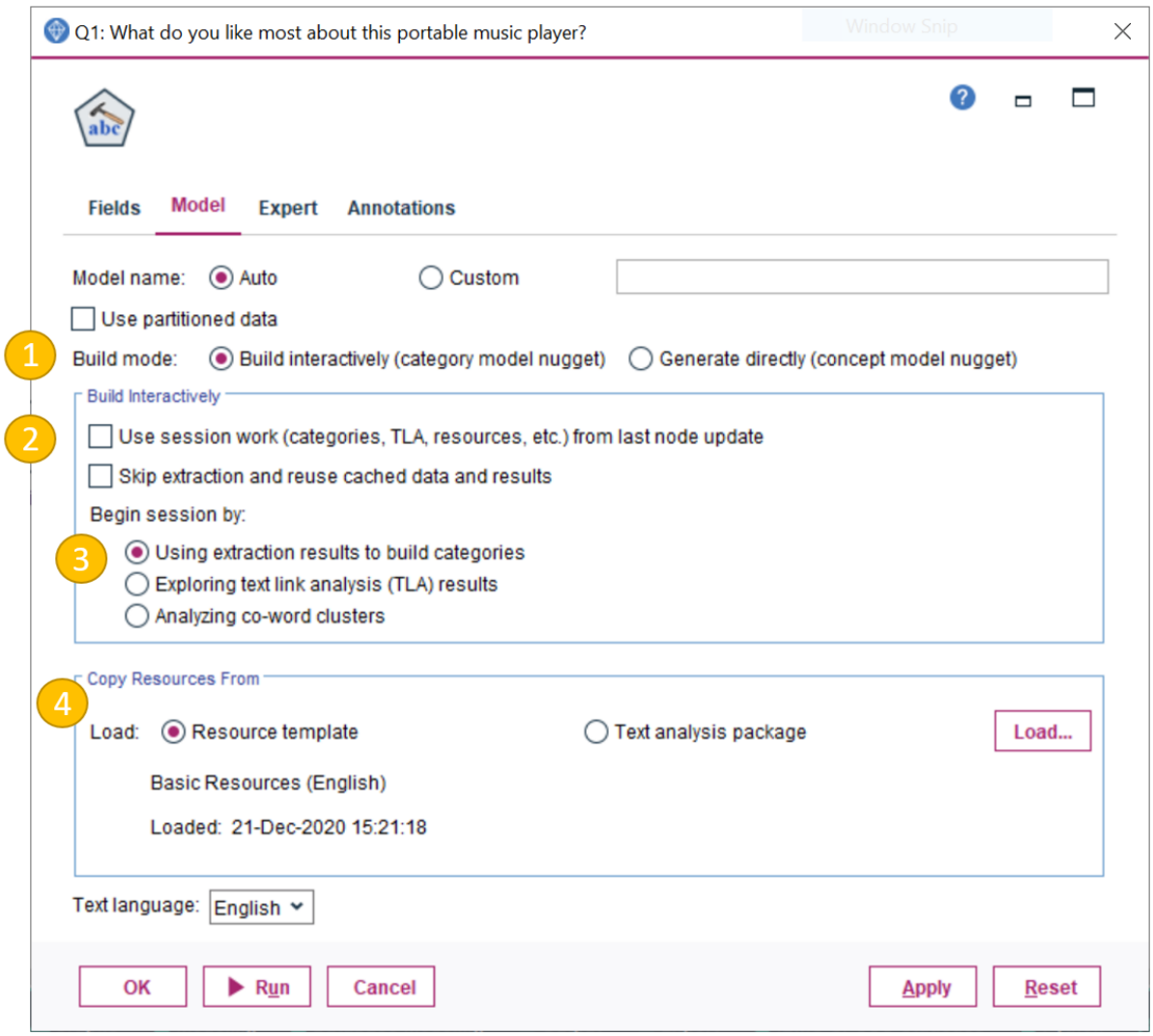


Figure 4.4 Model tab in the Text Mining node

The currently displayed **Model** tab contains the following controls:

1. **Build mode**:
    a. **Build Interactively:** Interactive mode opens up the interactive workbench and allows the user to view the extraction process, control the typing of terms and influence the model building process.
    b. **Generate directly:** this option directly creates a text mining model based on the concepts extracted. The option offers controls over the number concepts that are included in the model and whether or not the generated model should be based on the most frequently occurring concepts.
2. **Session work**:
    a. **Use session work**: checking this box enables users to continue using the extraction settings, categories, resources, and any other work from

a previous session when the node was updated. The option prevents the user from choosing a new resource template.

b. **Skip extraction and reuse cached data and results:** checking this box stops the extraction process entirely. It is used when the previous session was closed in such a way that not just the settings, but the data and the extraction results were cached in the Text Mining node for re-use, so that the user could continue from where they left off.

3. **Begin session by:**
   a. **Using extraction results to build categories**: this choice performs an extraction and opens the interactive workbench showing the extracted concepts.
   b. **Exploring text link analysis (TLA) results:** instead of performing just the extraction, this option identifies concepts and types that co-occur in the same response or document. This option requires that a resource template containing TLA pattern rules is selected or that the user is re-invoking a session where pattern rules have already been defined.
   c. **Analyzing co-word clusters**: This enables a clustering algorithm that creates groups of concepts based on the strength of a link value between them. The clusters themselves can be used to create categories. This mode of analysis is not covered by this course.
4. **Resource selection**:
   a. **Resource template**: Resource templates are collections of files that include libraries, compiled resources, and advanced linguistic resources. The templates are designed for those applying text mining to application areas like CRM, customer satisfaction or product satisfaction or for people working in industries such as insurance, banking, biosciences or IT. By loading an appropriate template, the user is effectively given a head start as many of the terms, acronyms and concepts associated with the area of interest will already have been defined within the resource template. Figure 4.5 shows the list of available English language resources that users can access when clicking the **Load** button.
   b. **Text analysis package**: Text analysis packages (or *.tap* files) are the easiest way to save an entire text mining project consisting of all the linguistic resources in a resource template (including types, synonyms, and excluded terms) as well as any defined categories or rules.

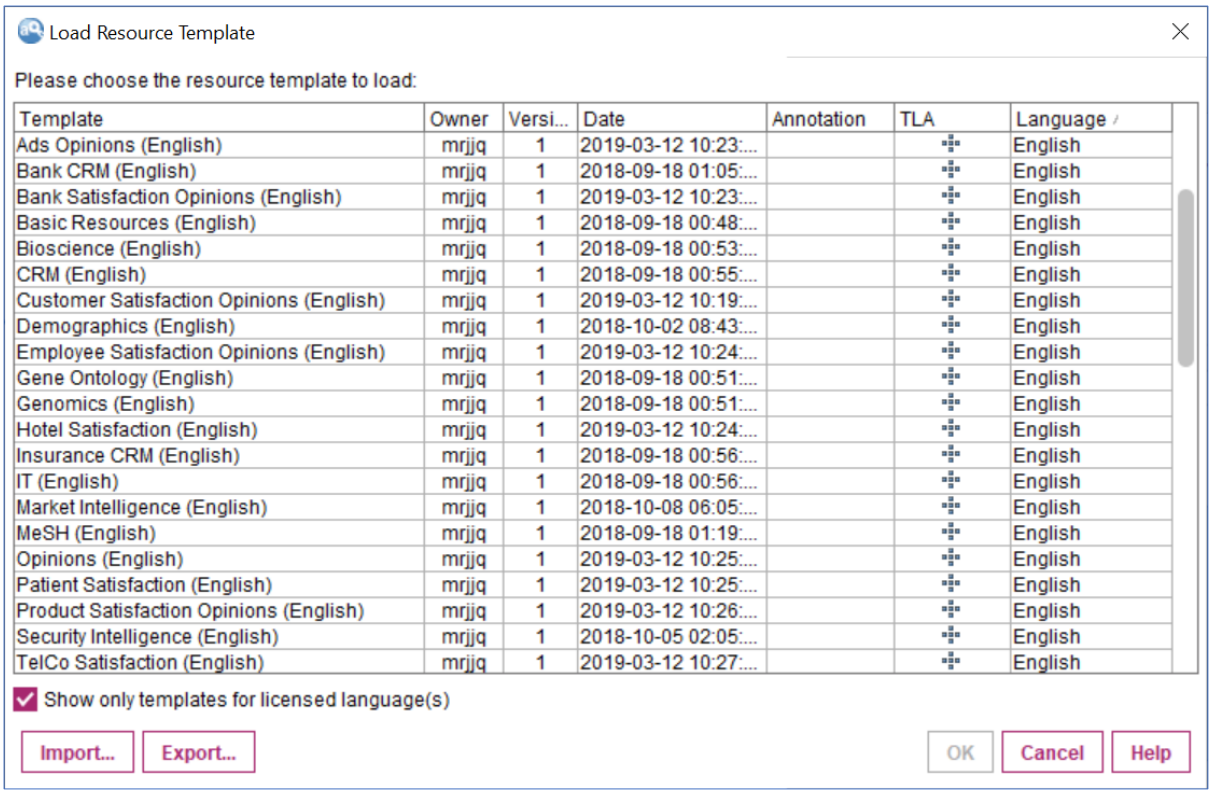


Figure 4.5 The Load Resource Template dialog showing Resource Templates available in IBM SPSS Modeler 18.2.1

Clicking on the **Expert** tab reveals access to a number of further settings controlling the extraction process. Figure 4.6 shows an image of the Expert control tab with numbered annotations.

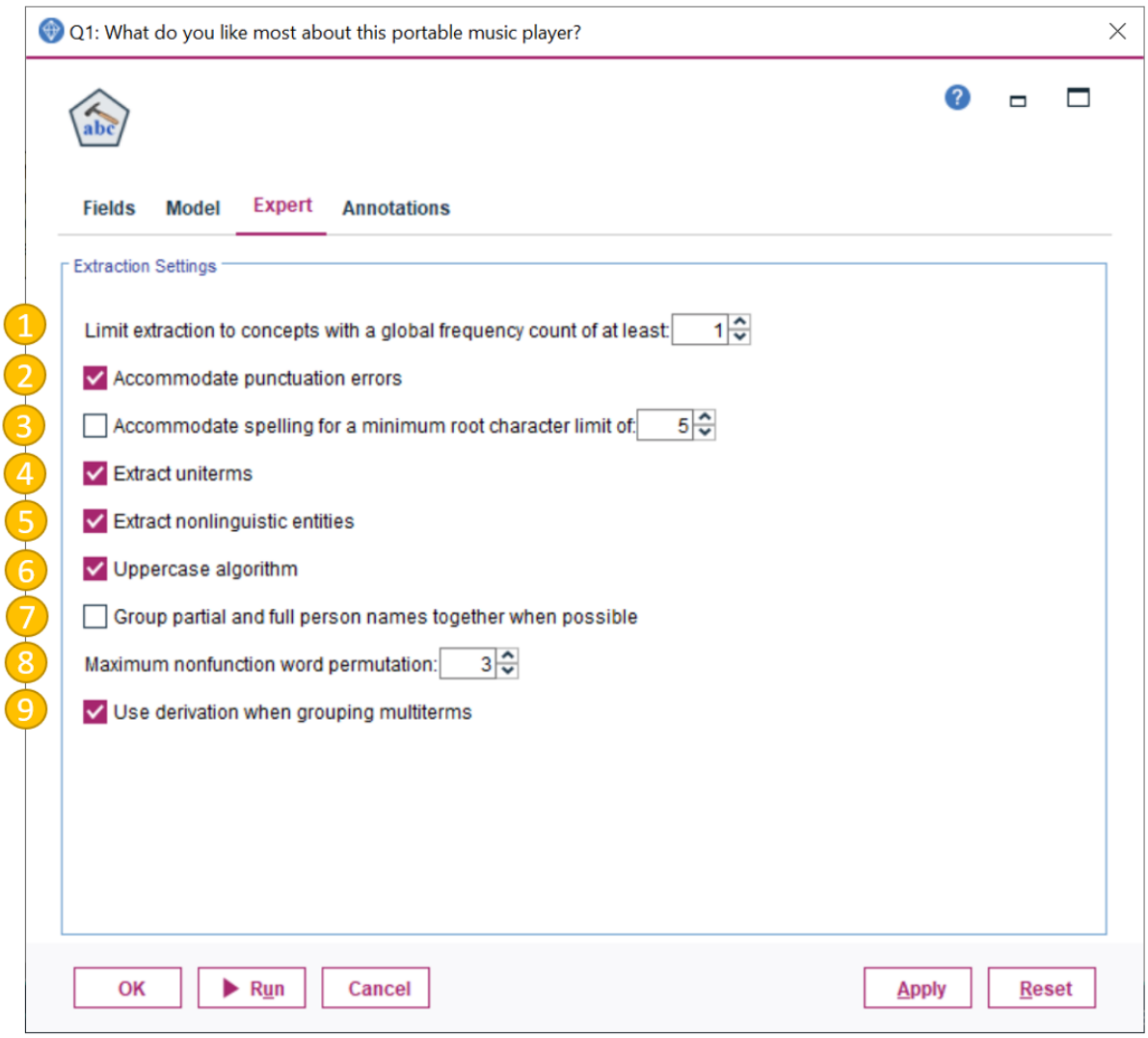


Figure 4.6 The Expert tab in the Text Mining node

The currently displayed **Expert** tab contains the following controls:

1. **Limit extraction to concepts with a global frequency of at least**: This specifies the minimum number of times a term or phrase must occur in order to be extracted. A value of 3 would mean only words or phrases that appeared at least 3 times in the corpus of text data would be extracted. Increasing this limit can have a noticeable effect on the number of compound terms that are extracted. Keeping the default value of 1 means that the extraction might find terms such as **manager** (3), **night manager** (1), **senior manager** (1) and **department manager** (1). However, increasing the value to 2 means that the compound terms would no longer be extracted as each one only has a frequency of 1. Instead, the user would see the concept **manager** occurring 6 times. The exception to this, is when the resource template includes certain compound terms that occur in the text irrespective of their frequency. For

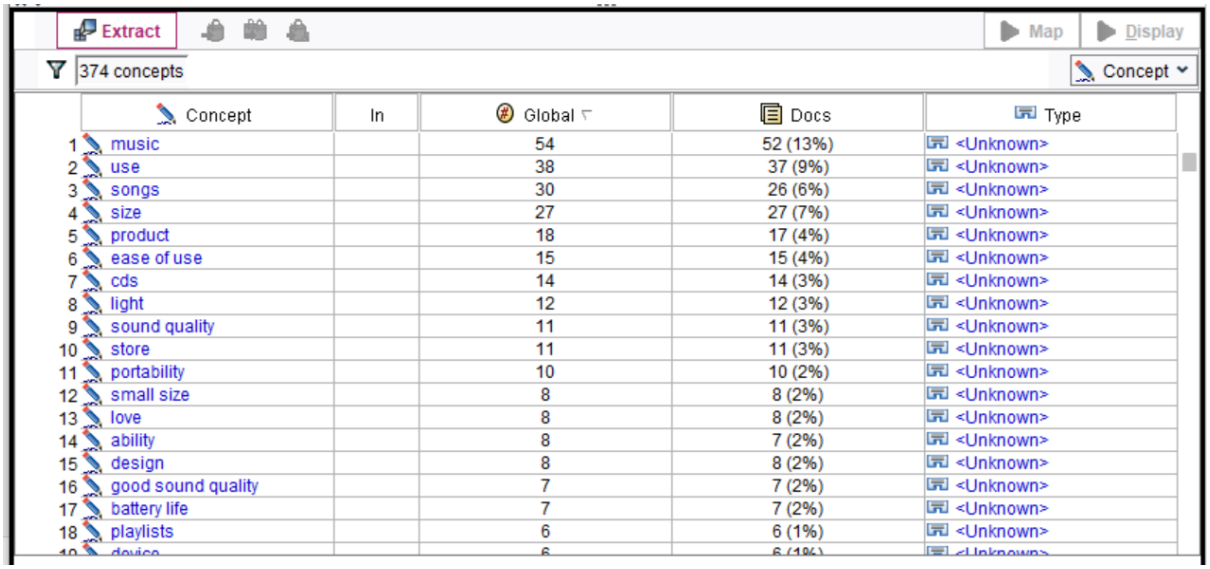
many projects, it's worth experimenting with this setting to see the effect of increasing the value above 1.

2. **Accommodate punctuation errors:** This option applies a normalization algorithm to the text to remove punctuation errors. These errors might include the incorrect placement of periods, commas, semicolons, and forward slashes. This temporary normalization is applied to improve the extractability of concepts. The option is particularly useful when dealing with short, abbreviated texts originating from call centre notes, CRM systems or open-ended surveys.
3. **Accommodate spelling for a minimum root character limit of:** Earlier we looked at the role of equivalence classes in the extraction process and referred to the spell-checking routine that the extraction system employs. It works by temporarily removing all vowels and multiple consonants from candidate terms before comparing them to see if they are the same word. This check-box option allows us to invoke the routine, but we can also set a minimum word size based on the number root characters. The number of root characters in a term is calculated by summing all of the characters and subtracting any letters that form inflection suffixes and, in the case of compound-word terms, determiners and prepositions. For example, removing the plural inflection from the term **invoices** means that there are 7 root characters as they form the word **invoice**. In a similar vein, the compound term **head of the organisation** would be counted as 16 root characters (**head organisation**) as the middle two words are ignored. This minimum threshold option therefore acts a filter controlling whether or not the fuzzy grouping spell-checking algorithm should be applied, as smaller words are more likely to be confused with other terms.
4. **Extract uniterms:** A **uniterm** is the technical name given to single words. Single words are extracted as long as they are not already part of a compound term and meet the parts of speech pattern criteria for extraction e.g., the word is either a noun or an unrecognized part of speech.
5. **Extract nonlinguistic entities:** Often text data contains information that is not part of the standard lexicon of linguistics. Data such as phone numbers, currency amounts, national insurance numbers, time values, dates, percentages, e-mails and website URLS are termed **nonlinguistic entities** and you can decide include or exclude these data from the extraction process. Users can also define new nonlinguistic entities in the configuration section of the **Advanced Resources** tab of the interactive workbench.
6. **Uppercase algorithm:** This option allows the extraction of uniterms and compound terms that are not recognised by the resource libraries as long as the first letter of the term is in uppercase. This setting offers a useful method to extract otherwise unknown nouns.

7. **Group partial and full person names together when possible:** Often text will use a mixture of a person's full name and their surname depending on the context. Switching on this option means that the extraction process will attempt to match any unknown uniterms, like **mcconnell** with a compound term such as **john mcconnell** that has already been identified as a person.
8. **Maximum nonfunction word permutation:** This is another equivalence class algorithm that tries to identify terms that relate to the same underlying concept. Setting the limit to 2 means that phrases such as **Berni is company CEO** and **Berni is CEO of the company** results in the single concept **company ceo** being extracted as there are only two non-functioning words i.e., **of** and **the**.
9. **Use derivation when grouping multiterms:** This final equivalence class algorithm uses the idea of concept derivation to group multiterms that mean the same thing. For example, when using the Customer Satisfaction Opinions resource template, the multiterms **staff communications** and **employee communication** are extracted as the single concept **staff communication** as **employee** is derivationally linked to **staff**.

4.2 Filtering extraction results

So far in this chapter, we have looked at the many optional elements that control the extraction process in the Text Mining node. Figure 4.7 shows the initial results in the extraction pane of the interactive workbench when we run the stream **04_The_Text_Mining_Node.str** using just the default settings. Here we can see the extracted concepts from the field **Q1: What do you like most about this portable music player?**.



	Concept	In	Global	Docs	Type
1	music		54	52 (13%)	<Unknown>
2	use		38	37 (9%)	<Unknown>
3	songs		30	26 (6%)	<Unknown>
4	size		27	27 (7%)	<Unknown>
5	product		18	17 (4%)	<Unknown>
6	ease of use		15	15 (4%)	<Unknown>
7	cds		14	14 (3%)	<Unknown>
8	light		12	12 (3%)	<Unknown>
9	sound quality		11	11 (3%)	<Unknown>
10	store		11	11 (3%)	<Unknown>
11	portability		10	10 (2%)	<Unknown>
12	small size		8	8 (2%)	<Unknown>
13	love		8	8 (2%)	<Unknown>
14	ability		8	7 (2%)	<Unknown>
15	design		8	8 (2%)	<Unknown>
16	good sound quality		7	7 (2%)	<Unknown>
17	battery life		7	7 (2%)	<Unknown>
18	playlists		6	6 (1%)	<Unknown>

Figure 4.7 Extracted concepts with default settings

In this instance the stream is using the Basic Resources (English) resource template and that an extraction based on the default settings has resulted in 374 concepts being extracted.

We can see that the most common concept is **music** which has occurred 54 times in the dataset (as indicated by the **Global** column) across 52 records (as indicated by the **Docs** column). Moreover, the Basic Resources template does not have a defined term type for this concept, so it appears as **Unknown** in the **Type** column. You can also see that the extracted concepts consist of single word **uniterms** as well as **compound terms** like **ease of use** and **sound quality**.

Figure 4.8 shows where to find the **Extraction Filter** dialog button .

This is a simple but important function within the extraction pane that allows us to focus on certain kinds of extracted concepts. In this example we only have 374 concepts, but with some data files, we might be dealing with thousands so it's important that users have control over the number of concepts shown and the ability to search for particular extracted concepts.

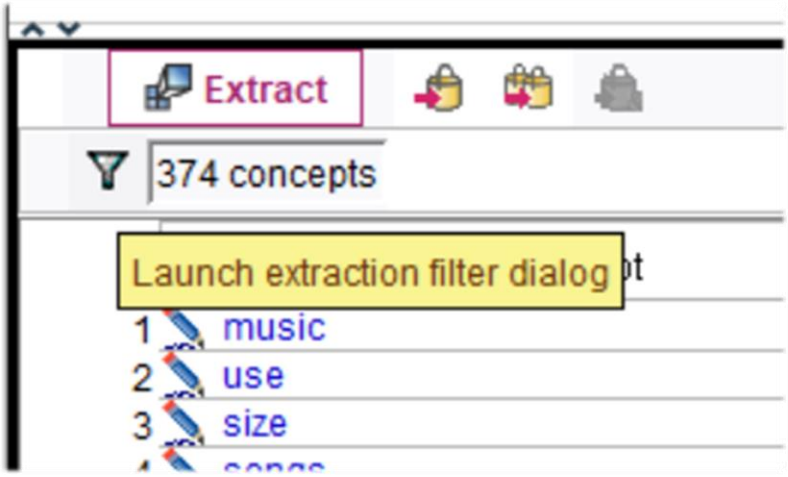
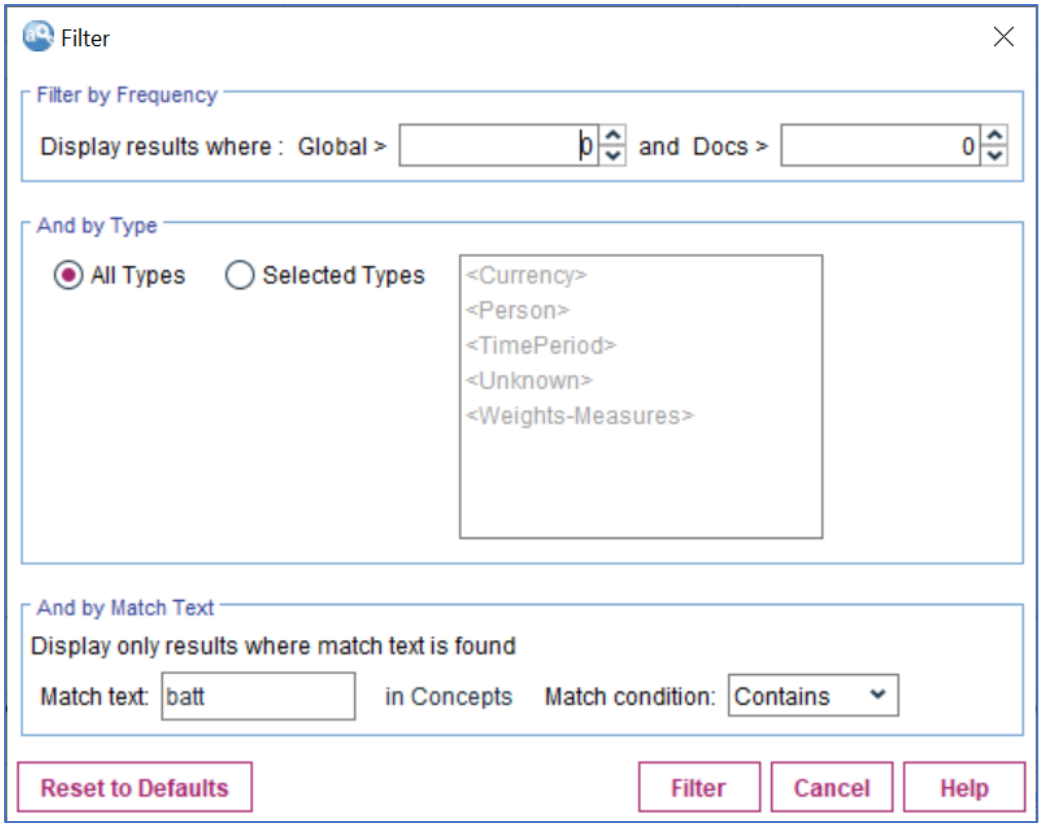


Figure 4.8 The Filter dialog button

The Filter dialog allows the user to:

- Limit the display of extracted concepts based on a minimum frequency of occurrence across the entire dataset (Global) or the minimum frequency of records (Documents) that they occur in
- Limit the display to concepts that only occur in certain term types
- Limit the display to concepts that match certain text strings

Figure 4.9 shows how the Extraction Filter dialog can be used to find concepts that contain the text string **batt**. Note that a binoculars icon appears next to the information box indicating that five concepts matched the condition.



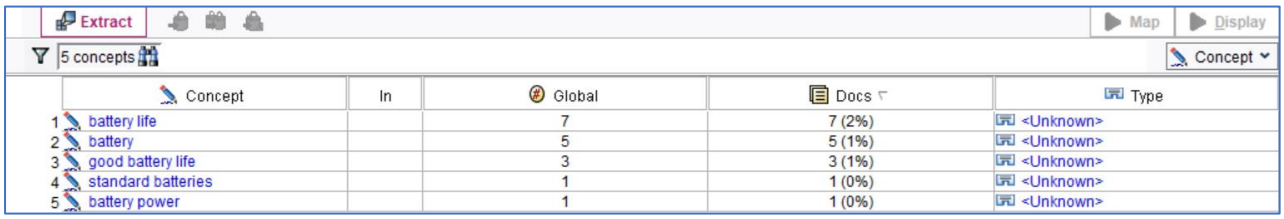


Figure 4.9 The Filter dialog used to match concepts containing the text string 'batt' and the resultant matching concepts displayed in the extraction pane

As Figure 4.10 shows, we can return to the Extraction Filter dialog and request that the extraction pane no longer displays concepts that belong to the **unknown** term type simply by selecting some of the other available types.

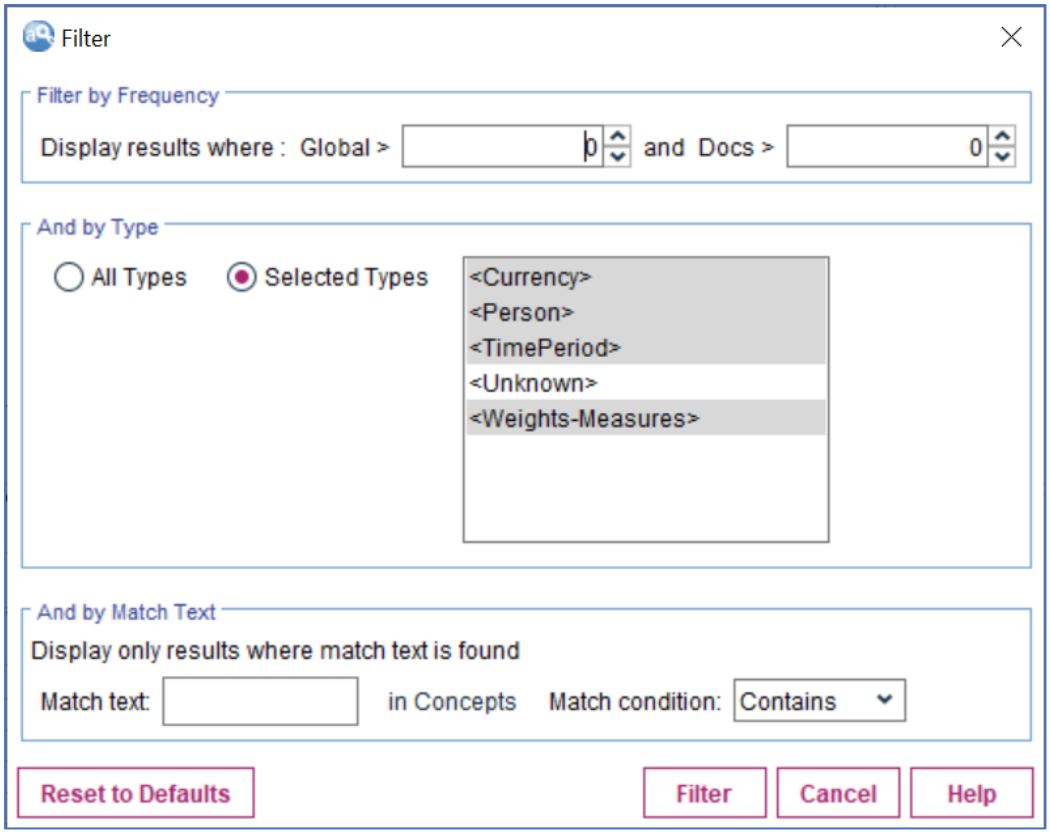
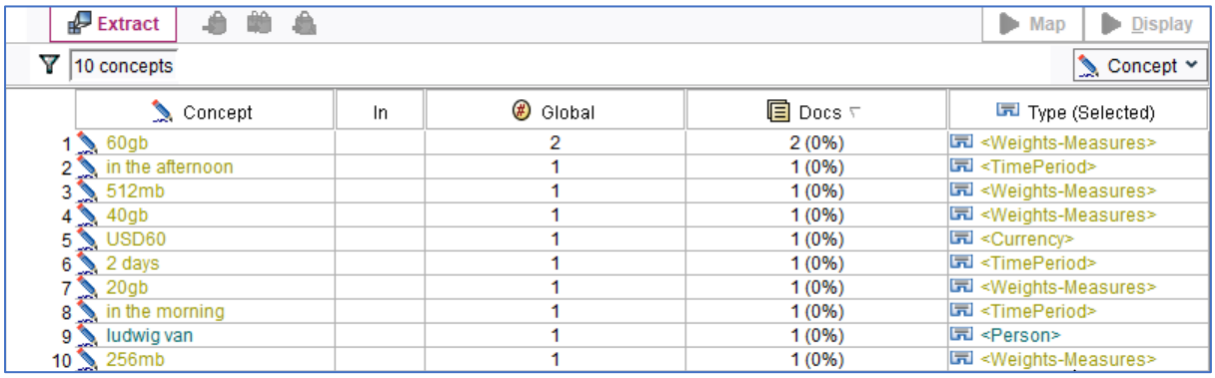


Figure 4.10 Using the Filter dialog to request that concepts are displayed in the extraction pane based on their type

As Figure 4.11 shows, only the concepts related to currencies, persons, time periods, and weights and measures are now displayed in the extraction pane. This is a good way for users to check that terms have been correctly assigned to their relevant type.



	Concept	In	Global	Docs	Type (Selected)
1	60gb		2	2 (0%)	<Weights-Measures>
2	in the afternoon		1	1 (0%)	<TimePeriod>
3	512mb		1	1 (0%)	<Weights-Measures>
4	40gb		1	1 (0%)	<Weights-Measures>
5	USD60		1	1 (0%)	<Currency>
6	2 days		1	1 (0%)	<TimePeriod>
7	20gb		1	1 (0%)	<Weights-Measures>
8	in the morning		1	1 (0%)	<TimePeriod>
9	ludwig van		1	1 (0%)	<Person>
10	256mb		1	1 (0%)	<Weights-Measures>

Figure 4.11 The result of using the Filter dialog to control which term types are displayed in the extraction panel

Having reset the filters to the default settings, within the extraction pane we can also switch the displayed results between concepts and types.

Clicking the following button in the top right-hand corner of the pane allows to change this display:

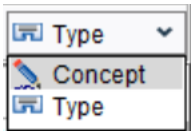
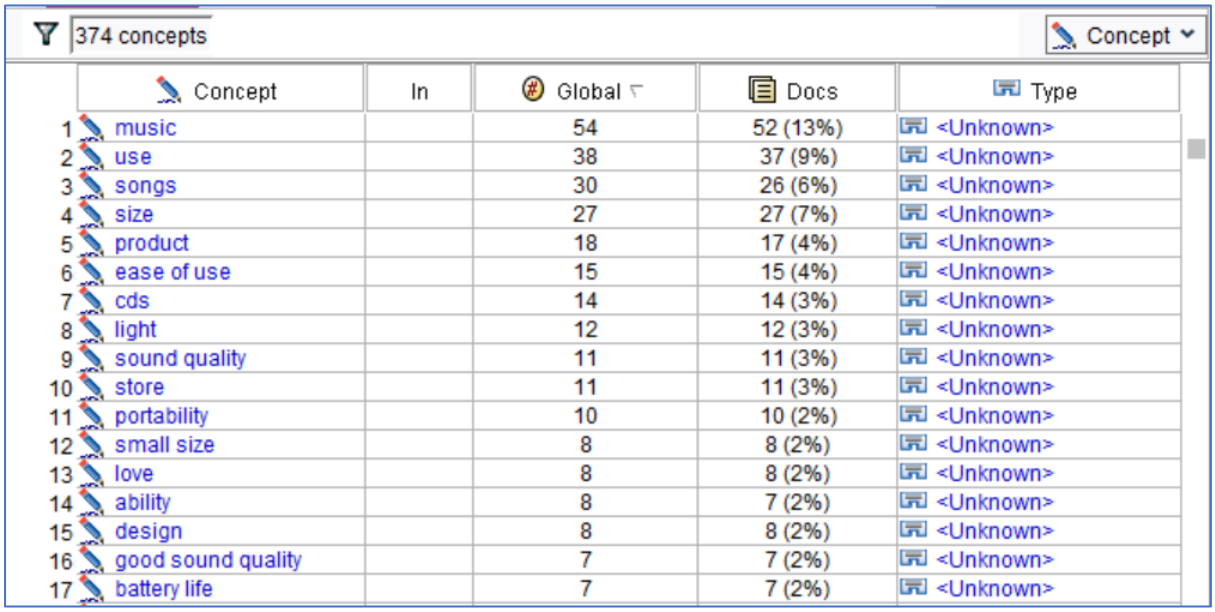


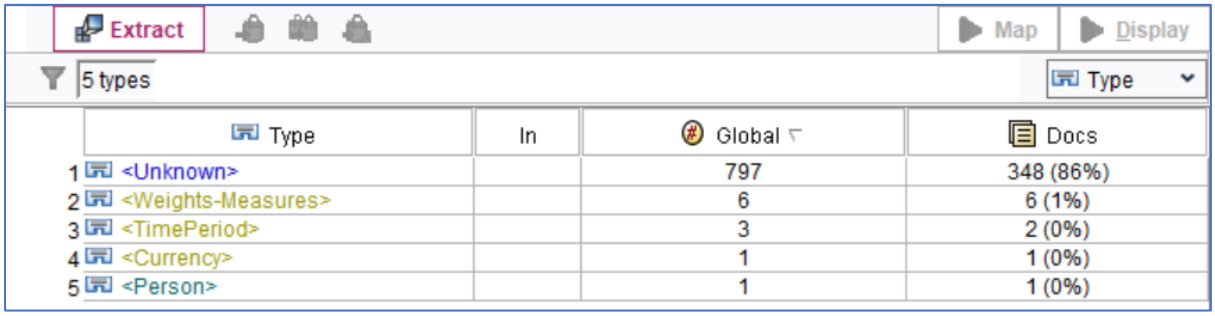
Figure 4.12 shows the default view of the extraction pane with extracted concepts displayed.



	Concept	In	Global	Docs	Type
1	music		54	52 (13%)	<Unknown>
2	use		38	37 (9%)	<Unknown>
3	songs		30	26 (6%)	<Unknown>
4	size		27	27 (7%)	<Unknown>
5	product		18	17 (4%)	<Unknown>
6	ease of use		15	15 (4%)	<Unknown>
7	cds		14	14 (3%)	<Unknown>
8	light		12	12 (3%)	<Unknown>
9	sound quality		11	11 (3%)	<Unknown>
10	store		11	11 (3%)	<Unknown>
11	portability		10	10 (2%)	<Unknown>
12	small size		8	8 (2%)	<Unknown>
13	love		8	8 (2%)	<Unknown>
14	ability		8	7 (2%)	<Unknown>
15	design		8	8 (2%)	<Unknown>
16	good sound quality		7	7 (2%)	<Unknown>
17	battery life		7	7 (2%)	<Unknown>

Figure 4.12 Individual concepts displayed in the extraction pane

Figure 4.13 shows the same pane, but this time the display has been switched to show the term types that the extracted concepts fall into.



	Type	In	Global	Docs
1	<Unknown>		797	348 (86%)
2	<Weights-Measures>		6	6 (1%)
3	<TimePeriod>		3	2 (0%)
4	<Currency>		1	1 (0%)
5	<Person>		1	1 (0%)

Figure 4.13 Term types displayed in the extraction pane

This simple approach to managing which extracted types and concepts are displayed in the extraction pane is invaluable, as it not only allows users to check that concepts

have been correctly extracted and typed, but also helps with the categorisation process as both terms and types can be dragged directly from this pane and added to existing categories or used to create new categories. For example, the filtered terms shown earlier in Figure 4.9, could all be added to a new category called **Battery** even though the word **battery** occurs as a uniterm on its own as well as part of several compound terms. In a similar vein, the **Weights-Measures** type could be added to a new category called **Capacity** as it only contains terms related to storage capacity such as **20gb**.

4.3 Changing the extraction settings

Before we finish this chapter, let's take a brief look at the effect of changing the default extraction settings. To do so, we can return to the Text Mining node and select the tab marked:

Expert

Within the Expert tab, uncheck the box marked:

Extract uniterms

To re-run the extraction, click:

Run

Figure 4.14 shows the results in the extraction pane.

203 concepts — Concept

	Concept	In	Global	Docs	Type
1	product		18	17 (4%)	<Unknown>
2	ease of use		15	15 (4%)	<Unknown>
3	cds		13	13 (3%)	<Unknown>
4	sound quality		11	11 (3%)	<Unknown>
5	small size		8	8 (2%)	<Unknown>
6	good sound quality		7	7 (2%)	<Unknown>
7	battery life		7	7 (2%)	<Unknown>
8	good sound		5	5 (1%)	<Unknown>
9	storage capacity		5	5 (1%)	<Unknown>
10	large capacity		4	4 (1%)	<Unknown>
11	download music		4	4 (1%)	<Unknown>
12	convenience		3	3 (1%)	<Unknown>
13	size		3	3 (1%)	<Unknown>
14	great sound		3	3 (1%)	<Unknown>
15	good battery life		3	3 (1%)	<Unknown>
16	large storage		3	3 (1%)	<Unknown>
17	light weight		3	3 (1%)	<Unknown>

Figure 4.14 Extraction results with 'Extract uniterms' switched off

Although the extraction pane now only shows 203 concepts, it does appear to display a larger proportion of compound terms than previously. However, there are still

some single words (or uniterms) such as **product** and **cds** showing. This is because the extractor's uppercase algorithm is still in use, whereby unknown terms that begin with a capital letter are extracted. Figure 4.15 shows that when we click on a single concept in the extraction pane, the accompanying **Data pane** highlights the associated term as it appears in a text snippet. You can see that the concept **product** is associated with mentions of **Product A**.

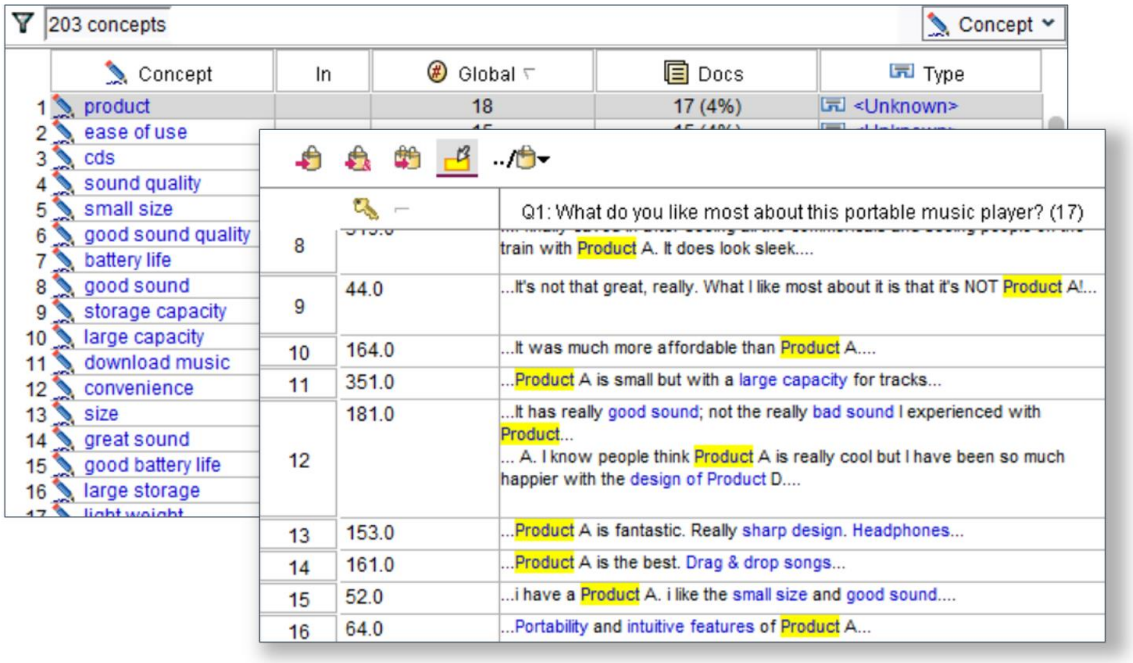


Figure 4.15 Mentions of the uniterm 'product' as it appears in the Data pane

You may recall however, that we can switch off the uppercase algorithm in the extraction settings. To do so, simply return to the Text Mining node and select the tab marked:

Expert

Within the Expert tab, uncheck the box marked:

Uppercase algorithm

And re-run the extraction by clicking:

Run

Figure 4.16 shows the results.

178 concepts — Concept

	Concept	In	Global	Docs	Type
1	ease of use		15	15 (4%)	<Unknown>
2	sound quality		11	11 (3%)	<Unknown>
3	small size		8	8 (2%)	<Unknown>
4	good sound quality		7	7 (2%)	<Unknown>
5	battery life		7	7 (2%)	<Unknown>
6	good sound		5	5 (1%)	<Unknown>
7	storage capacity		5	5 (1%)	<Unknown>
8	large capacity		4	4 (1%)	<Unknown>
9	download music		4	4 (1%)	<Unknown>
10	large storage capacity		3	3 (1%)	<Unknown>
11	great sound		3	3 (1%)	<Unknown>
12	light weight		3	3 (1%)	<Unknown>
13	fm radio		3	3 (1%)	<Unknown>
14	good battery life		3	3 (1%)	<Unknown>
15	great sound quality		2	2 (0%)	<Unknown>

Figure 4.16 Extraction results with 'Extract uniterms' *and* 'Uppercase algorithm' switched off

The extraction pane now shows that there are no uniterms in among the unknown concepts. We should bear in mind that having ran the extraction process three times, we now have three copies of the interactive workbench open simultaneously. Figure 4.17 shows the contents of the **Outputs** tab in IBM SPSS Modeler.

Figure 4.17 Outputs tab showing three copies of the interactive workbench open

If we no longer need these instances of the workbench, we can free up valuable memory resources by closing them. To close each of them, within the Outputs tab:

Shift-click to select each instance of interactive workbench

Right-click and select:

Delete

Then click:

Continue

Practice Exercise – Chapter 4

Within the folder **Student Exercises** open the following stream:

Chapter_04_Practice.str

1. Run the first text mining branch labelled **Concepts with a global frequency of at least 1.** Click **OK** at the warning dialog.

2. Run the second text mining branch labelled **Concepts with a global frequency of at least 3.** Click **OK** at the warning dialog.

3. You will now have two interactive sessions open. Notice that in the first session, the term **cost** has a global frequency of **26**, whereas in second it has a global frequency of **30**. Why should this be?

As a clue, in the **first** session, use the filter dialog  to match concepts containing the term **cost.**

Close the and **Exit** both sessions without updating the nodes.

4. From the Text Mining palette in Modeler, attach a new Text Mining node to the data source node. Edit the node in the following way:

- Specify the ID field (optional) to be **ID**
- Specify the text field to be **Q1leisurefactors**
- In the **Model** tab, load the **Opinions (English)** resource template
- In the **Expert** tab, switch on the setting **Accommodate spelling for a minimum root character limit of 5**

Now run the node.

5. Switch the view of the extraction pane so that it shows **Type** groups rather than extracted **Concepts**

6. Switch the view back to **Concepts** and use the filter controls so that only **Positive** concepts are shown. Select the concept **Excellent** and click the **Display** button to see the underlying terms. Note that most of these terms and phrases are synonyms of the word **Excellent**. Can you find other examples of extracted concepts where the terms in the data are not exactly the same word?

7. Now use the filter controls so that only **Unknown** concepts are shown. Are all of these concepts useful or can some be ignored? If we were to begin creating a project, which concepts might deserve their own type groups?

8. Return to the filter controls and switch off the current filter so that concepts from all the type groups are displayed.

`Close` the and **`Exit`** the session without updating the node.

Chapter 5 Defining Types

Earlier in the course we introduced the idea of term types. In this chapter, we will explore how types are defined and applied to extracted concepts. Creating types is a powerful way to organise extracted concepts so that the text data may be more easily categorised. Figure 5.1 shows the relationship between types and concepts. It further shows that it's possible that an extracted concept also appears as a target term in the synonym dictionary. Users should be aware that sometimes a concept may be already linked to a target synonym, which in turn is already assigned to a type. In such cases, it may be necessary to delete the concept's link to the target synonym in order to assign it correctly to its relevant type group.

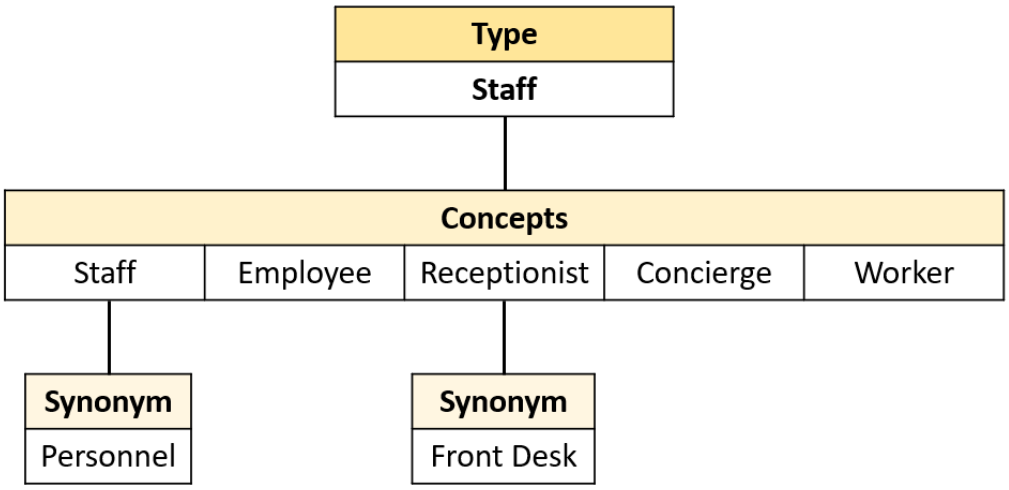


Figure 5.1 Relationship between concepts and types

Defining new types is usually an essential part of the text mining process. In doing so, users can build up entire dictionaries of concepts, synonyms and types that are specially customised to a particular application.

5.1 Matching concepts to types

One of the reasons that creating term types is so useful in Modeler Text Analytics, is that it can force the extraction of words that the parts of speech algorithm ignores. The exception to this, is when the word, or words, are already part of a compound phrase that has been extracted. It is however possible to override these situations as long as a word or phrase is assigned to a type using the correct matching method. Matching methods are used to ensure that the context in which a word or phrase appears is exploited to correctly assign it to its term type. Given that a word such as **training** might appear as part of a longer phrase such as **skills training**, **training provider** or **training for new staff**, a user might want to control how it is assigned to

a type based on whether it appears as a suffix, prefix or in the mid-section of a phrase. With this in mind, the following controls in Figure 5.2 are available to users when allocating concepts to types.

Match Type	Match Description
Entire Term	This is the default method. It's a relatively a strict approach because the extracted concept has to exactly match the term as it is defined in its Type Properties dialog.
Start	Only the first word in a concept extracted from the text is matched. For example, if you enter **milk**, the Start option means **milk chocolate** will be matched.
End	Only the last word in a concept extracted from the text is matched. For example, if you enter **milk**, the End option means **cold milk** will be matched.
Any	The term is matched wherever it appears in the phrase. For example, if you enter **milk**, the Any option will assign **milk chocolate**, **dairy milk** and **dairy milk chocolate** to the same type. Note: this *only* works if the term is part of a compound concept.
Start or End	The term is matched if it appears at the start *or* the end of the concept. For example, if you enter **milk**, the Start or End option will type **milk chocolate** or **dairy milk** the same way.
Entire and Start	Either the extracted text or the *first* word must match the extracted concept
Entire and End	Either the extracted text or the *last* word must match the extracted concept
Entire and Any	This option is used to match *single word* terms to a type, no matter where they occur in the text.
Entire and (Start or End)	Either the entire extracted text, the first word or the last word must match the term in the dictionary.
Entire (no compounds)	If any part of the extracted concept matches the target term, the type is assigned, and the extraction is stopped *to prohibit the extraction from matching the term to a longer compound*. For example, if you enter **milk**, the Entire (no compounds) option will type the word **milk** and not extract the compound **chocolate milk** unless it is forced to somewhere else.

Figure 5.2 Match controls for assigning concepts to types

We can explore the various type-matching methods using a dataset containing four simple phrases as shown in Figure 5.3. The same example can be viewed using the stream **05_Exploring_Match_Types.str**.

	Text
1	I hate milk
2	I prefer cold milk
3	I bought milk chocolate
4	I like cold milk chocolate

Figure 5.3 Text phrases containing the term 'milk'

Figure 5.4 shows which concepts the parts of speech algorithm extracts from these phrases when using the Basic Resources (English) resource template. Note that with this resource template, all of the concepts are assigned to the **Unknown** type.

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Unknown>
2	cold milk		1	1 (25%)	<Unknown>
3	milk		1	1 (25%)	<Unknown>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.4 Concepts extracted from simple dataset

Later, we will take a closer look at the various options associated with creating types, but for now let's assume we've created a new type group called **milk** and we have assigned a single term (also called **milk**) to it. The following figures show how changing the match type affects which of the extracted concepts are typed as **milk**.

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Unknown>
2	cold milk		1	1 (25%)	<Unknown>
3	milk		1	1 (25%)	<Milk>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.5 Match type: Entire Term

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Milk>
2	cold milk		1	1 (25%)	<Unknown>
3	milk		1	1 (25%)	<Unknown>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.6 Match type: Start

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Unknown>
2	cold milk		1	1 (25%)	<Milk>
3	milk		1	1 (25%)	<Unknown>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.7 Match type: End

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Milk>
2	cold milk		1	1 (25%)	<Milk>
3	milk		1	1 (25%)	<Unknown>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.8 Match type: Start or End

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Milk>
2	cold milk		1	1 (25%)	<Unknown>
3	milk		1	1 (25%)	<Milk>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.9 Match type: Entire or Start

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Unknown>
2	cold milk		1	1 (25%)	<Milk>
3	milk		1	1 (25%)	<Milk>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.10 Match type: Entire or End

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Milk>
2	cold milk		1	1 (25%)	<Milk>
3	milk		1	1 (25%)	<Milk>
4	cold milk chocolate		1	1 (25%)	<Milk>

Figure 5.11 Match type: Entire or Any

	Concept	In	Global	Docs	Type
1	milk chocolate		1	1 (25%)	<Milk>
2	cold milk		1	1 (25%)	<Milk>
3	milk		1	1 (25%)	<Milk>
4	cold milk chocolate		1	1 (25%)	<Unknown>

Figure 5.12 Match type: Entire Start or End

	Concept	In	Global	Docs	Type
1	milk		4	4 (100%)	<Milk>
2	cold		2	2 (50%)	<Unknown>
3	chocolate		2	2 (50%)	<Unknown>

Figure 5.13 Match type: Entire (No Compounds)

As Figure 5.13 shows the Entire (No Compounds) method simply extracts the term **milk** four times. Any compound terms containing this word are not extracted.

5.2 Defining new types

Now that we've explored how concepts are matched to types, we start to look at how new types are designed. Figure 5.14 shows a sample of the initial extraction results that are generated when we execute the stream **05_Definining_Types.str**. Bear in mind that once again we are using the Basic Resources (English) template so most of the extracted concepts are typed as **Unknown**.

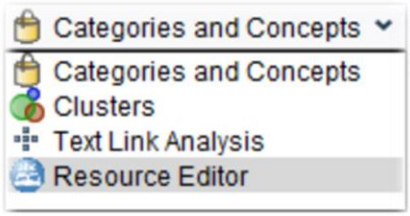
Extract | Map | Display

374 concepts | Concept

	Concept	In	Global	Docs	Type
1	music		54	52 (13%)	<Unknown>
2	use		38	37 (9%)	<Unknown>
3	songs		30	26 (6%)	<Unknown>
4	size		27	27 (7%)	<Unknown>
5	product		18	17 (4%)	<Unknown>
6	ease of use		15	15 (4%)	<Unknown>
7	cds		14	14 (3%)	<Unknown>
8	light		12	12 (3%)	<Unknown>
9	sound quality		11	11 (3%)	<Unknown>
10	store		11	11 (3%)	<Unknown>
11	portability		10	10 (2%)	<Unknown>
12	small size		8	8 (2%)	<Unknown>
13	love		8	8 (2%)	<Unknown>
14	ability		8	7 (2%)	<Unknown>
15	design		8	8 (2%)	<Unknown>
16	good sound quality		7	7 (2%)	<Unknown>
17	battery life		7	7 (2%)	<Unknown>
18	playlists		6	6 (1%)	<Unknown>
19	device		6	6 (1%)	<Unknown>
20	fact		6	5 (1%)	<Unknown>
21	capacity		6	6 (1%)	<Unknown>
22	battery		5	5 (1%)	<Unknown>
23	video		5	5 (1%)	<Unknown>
24	good sound		5	5 (1%)	<Unknown>
25	software		5	5 (1%)	<Unknown>
26	style		5	5 (1%)	<Unknown>

Figure 5.14 Initial extraction results from MP3 player survey

To define a new term type from scratch, we need to access the **resource editor** within the interactive workbench. To do so, click the drop-down menu in the top right-hand corner of the workbench:



From the menu, select:

Resource Editor

The workbench now switches to a view of the resource editor window that we can see in Figure 5.15.

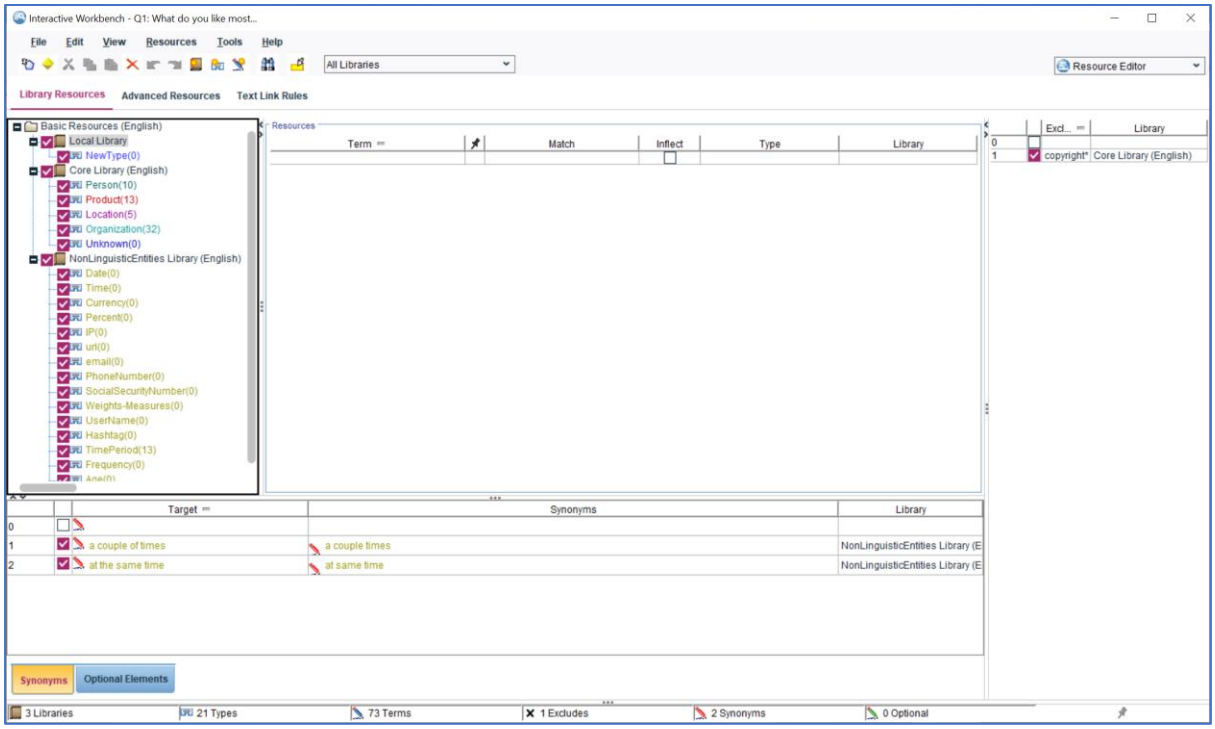


Figure 5.15 The resource editor using the Basic Resources (English) template

Later, we will take a closer look at the resource editor, but for now we need only to understand that this is where users of Modeler Text Analytics can edit and enhance existing resource templates. The main uses of the resource editor include editing and defining the following aspects of the resource templates:

- **Types** – these can be changed within or added to new dictionaries.
- **Synonyms** – these can be added, edited or removed to enhance clarity.
- **Optional Elements** – these are terms used to group variants of words together. Examples of optional elements are terms like **inc**. or **corp**. Optional elements usually refer to business terms.

- **Excluded terms** – these are a collection of terms and types that are removed from extracted results (e.g., **copyright**).

The Basic Resources (English) template contains three libraries shown on the left side of the editor window. Each library contains a collection of types and associated terms. Figure 5.16 shows an image of these libraries with their folder contents expanded.

Figure 5.16 The default term type libraries in the Basic Resources (English) template

You may recall that the Core library for English resources already contains five pre-defined types (Person, Product, Location, Organization and Unknown). These are *compiled* resources where most of the contents of the supplied type information are not visible to the user. Indeed, Figure 5.17 shows that if we click on the type **Location** we can see that it seems to contain only a few terms in the associated editor window. This is because it's a pre-built compiled resource, and these terms are merely sample locations. In reality, the system can identify thousands of locations, but nevertheless we can add new ones to the existing resource if needed.

Term	Match	Inflect	Type	Library
county	End		Location	Core Library (English)
great britain	Entire (no compounds)		Location	Core Library (English)
las vegas	Entire (no compounds)		Location	Core Library (English)
los angeles	Entire (no compounds)		Location	Core Library (English)
new zealand	Entire (no compounds)		Location	Core Library (English)

Figure 5.17 Sample terms in the Location type as part of the compiled Core Library

It's also worth mentioning that when we start an interactive workbench session, Modeler Text Analytics actually loads a *copy* of any selected resource template. This means that any edits or changes to the resources are specific to that instance. As a result, any future projects will always begin with an unaltered copy of the original template.

For our purposes, in this chapter the most important library is the one marked **Local**. By default, any new interactive workbench session includes an initially empty local library, and it is here that we can create our own custom library of new types that are relevant to our project data.

To illustrate this, we will create a type called **music** that captures a number of associated terms. One way to do this is to:

Right-click on* NewType *within the* Local Library *folder

From the drop-down menu click:

New Type

Figure 5.18 shows this.

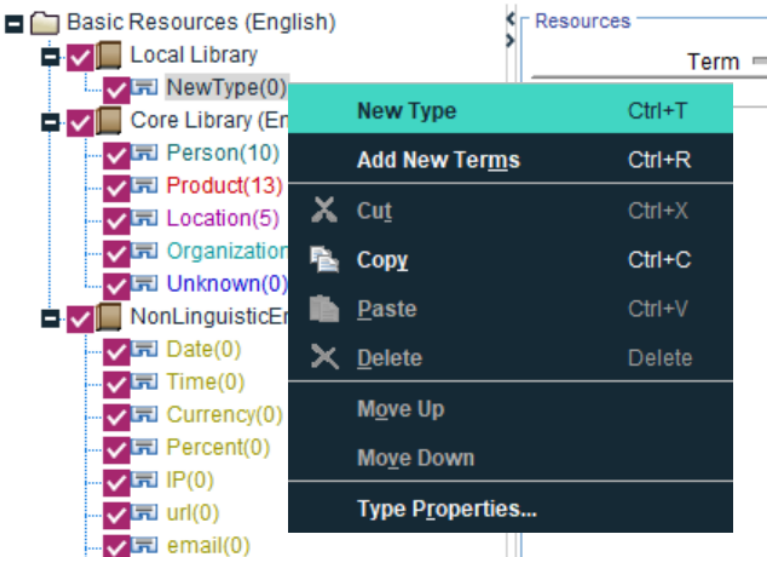


Figure 5.18 Creating a new type in the Local library

This action generates the **Type Properties** dialog where we can give our new term type a name as well as assign a match condition and custom colour. Figure 5.19 shows an annotated guide to this dialog.

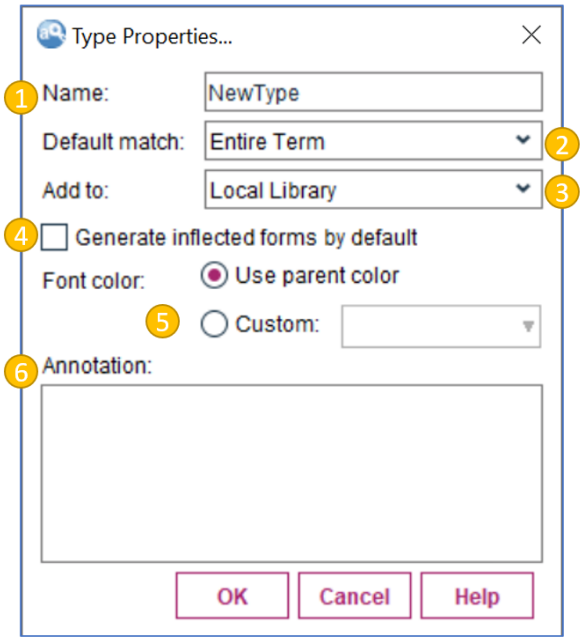


Figure 5.19 The Type Properties dialog

The Type Properties dialog allows us to control the following aspects of a new term type:

1. **Name**: Define a name for the new type. Remember that the type itself may need to contain a term with the same name.
2. **Default match**: This defines the default matching method for any terms associated with the type. It's important to note that the default method, **Entire Term**, is one of the most restrictive. If this is inappropriate, it's good practice to change this default before adding terms to the type category.
3. **Add to**: This allows the user to specify which library to add the new type to. There may be situations where more than one new library has been created by the user. As such it is not essential that new types are added only to the default local library.
4. **Generate inflected forms by default**: This checkbox option is particularly useful when dealing with nouns that may contain plural forms. It allows the user to add terms such as **child**, **mouse** or **leaf** and the extraction engine will still recognise the words **children**, **mice** and **leaves** as inflected forms of these terms.
5. **Font color**: Colour coding the types makes it much easier for users to navigate through the extraction results and spot any errors or overlooked terms that

should be added to existing or new types. Again, it is good practice to select an appropriate custom colour.

6. **Annotation**: Annotating new types is particularly helpful when new resources are created with a view to sharing them between users or projects, as it makes it easier for colleagues to understand the logic behind the type definition and what kind of terms it should capture.

To continue the process of creating a new type, assign the following values to the Type Properties dialog.

- **Name**: Music
- **Default match**: Entire and Any
- **Add to**: Local Library
- **Generate inflected forms by default**: Yes
- **Font color**: Orange
- **Annotation**: An example of creating a Type for terms related to music

The completed dialog is shown in Figure 5.20.

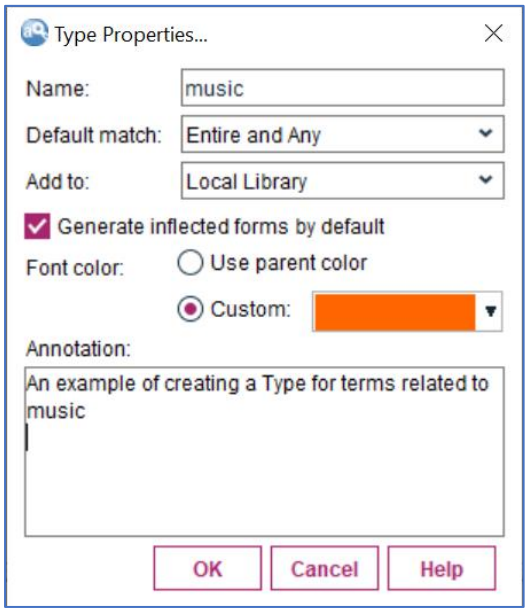


Figure 5.20 Completed Type Properties dialog for the new term type 'music'

At this point, we can enter terms to be associated with this type in the following way:

Right-click on the newly defined music term type

From the resultant drop-down menu choose:

Add New Terms

This process is shown in Figure 5.21.

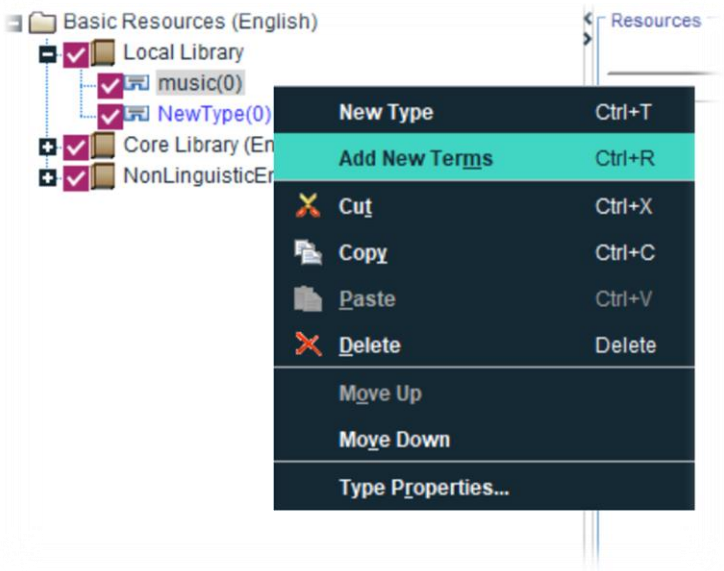


Figure 5.21 Adding new terms to the term type 'music'

The **Add New Terms** dialog appears. Enter the following terms separated by commas as shown in Figure 5.22 and click **OK**.

music

tune

song

track

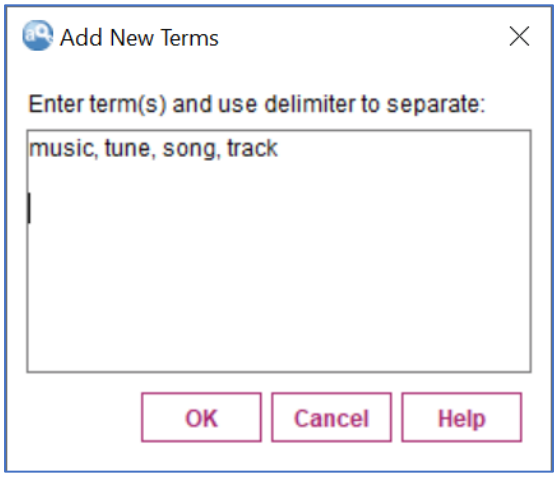


Figure 5.21 New terms added to the term type 'music'

Once these terms have been added, we can view them in the accompanying **Resources** pane for the term type **music** (see Figure 5.22).

Term	Match	Inflect	Type	Library
music	Entire and Any	☑	music	Local Library
tune	Entire and Any	☑	music	Local Library
song	Entire and Any	☑	music	Local Library
track	Entire and Any	☑	music	Local Library

Figure 5.22 Resources pane showing newly added terms for the term type music

You can see from the resources pane that each term has been assigned the match method **Entire and Any** and that the system may look for inflected versions of each one. You should note that it is possible to enter new terms directly into this window as well as using the **Add New Terms** dialog.

To see the effect of creating a new type, we need to return to the **categories and concepts** window by clicking on the drop-down menu in the top right-hand corner of the resource editor (see Figure 5.23).

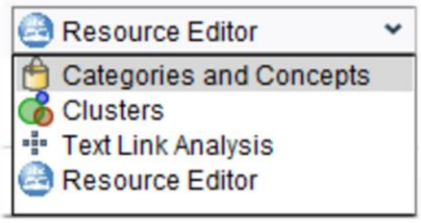


Figure 5.23 Switching back to the categories and concepts window

The first thing we notice when we return to the categories and concepts window is that the extraction pane is now shaded yellow (see Figure 5.24). This is alerting us to the fact that the typing of concepts occurs as part of the extraction process, so given that a new type has been added to the local library of the resource template, the extractor needs to be executed again to take account of it.

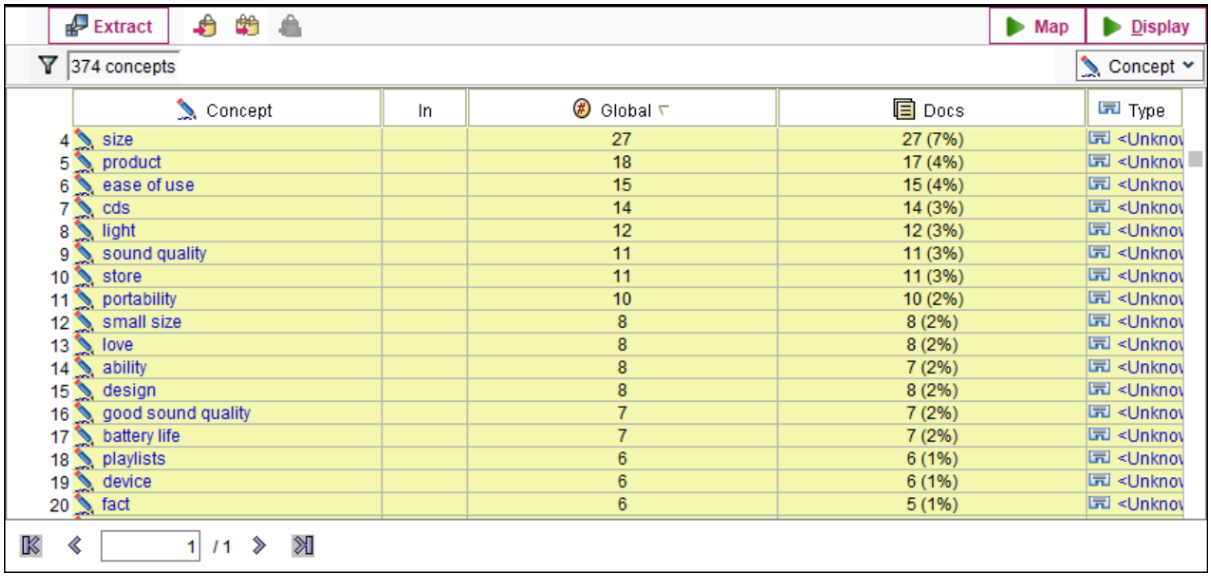


Figure 5.24 The extraction pane shaded yellow indicating that the data requires the extractor to be re-run

To re-run the extractor, click:

Extract

After the extraction process is re-executed, we can immediately see that a new type has appeared in the results. As Figure 5.25 shows, the concepts **music** and **song** are now matched with the term type **music**.

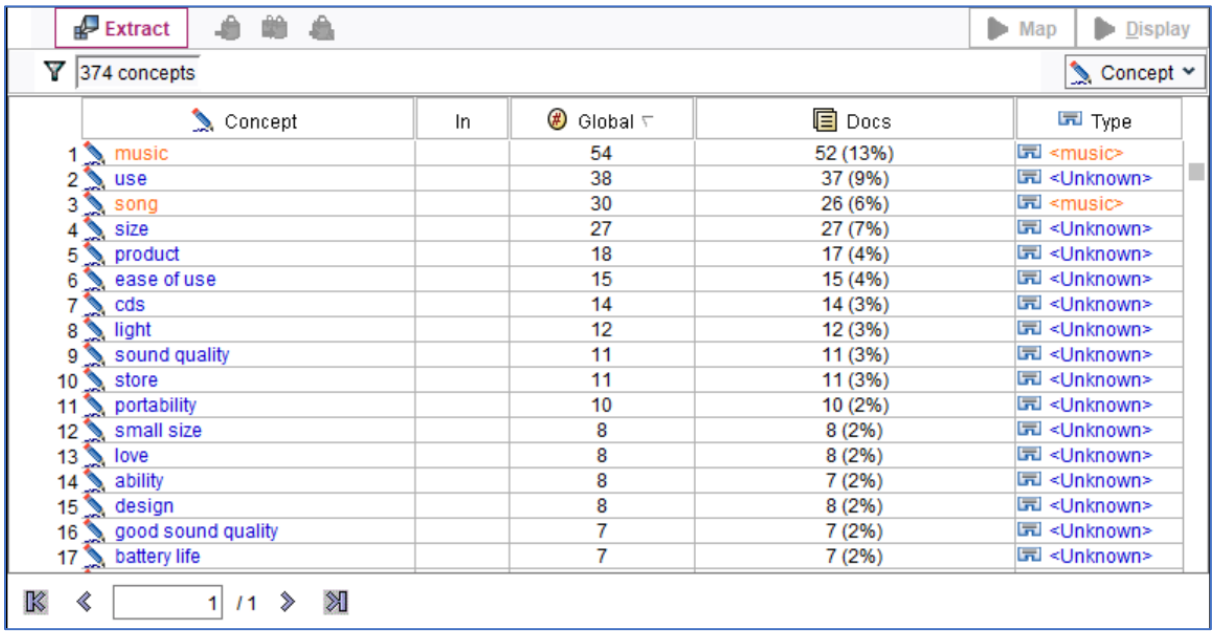


Figure 5.26 Concepts matched with the term type 'music'

Changing the extract pane to the type view, shows that there were 126 instances of concepts in 108 records that were matched with the new type group.

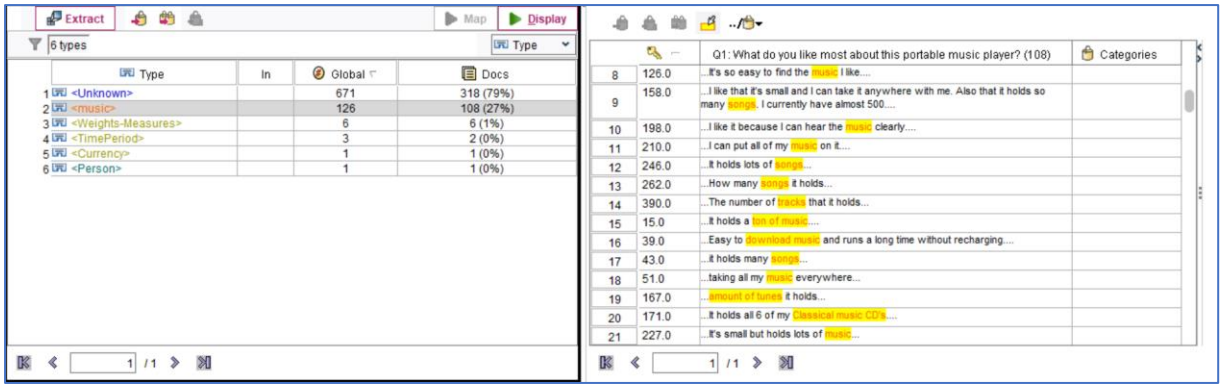


Figure 5.27 The extraction pane type list showing 126 concepts in 108 records matching with the type 'music'

When creating new term types, users are often trying to anticipate the words and phrases that might occur in the existing text (or future data sources) that should logically be associated with the newly created type. With that in mind, a common situation is that they may spot extracted concepts or phrases in the text that ought to be added to an existing type. In these situations, they can add a concept directly to an existing type from the extraction pane itself. In this dataset, it might be argued that, as many respondents mention their CDs, this specific term should *also* be added to the **music** type. Returning to the concept view of the extract pane, we can illustrate how to do this:

Right-click on the concept **CDs**

From the pop-up menu select:

Add to Type

From the sub-menu, click:

Music

This process is illustrated in Figure 5.28.

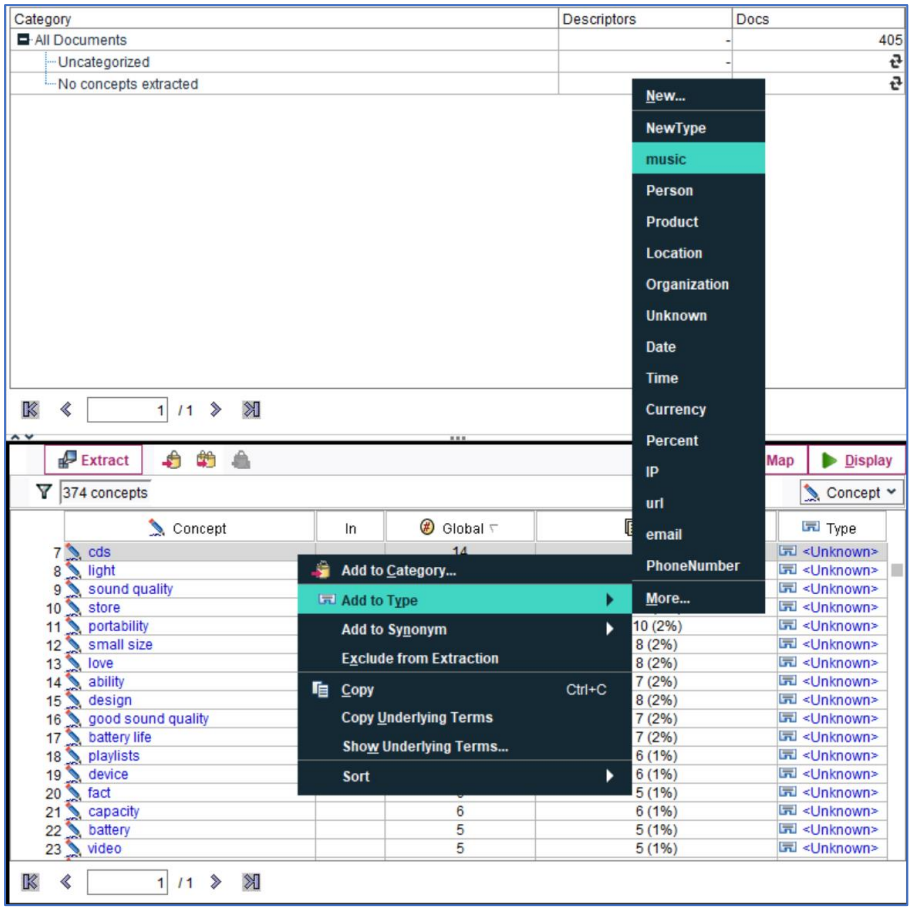


Figure 5.28 Adding a concept to an existing type from the extract pane

To see the effect of adding a new concept to the **music** type, the extraction process will need to be run again. Doing this every time a new concept needs to be added to a type can prove to be extremely time-consuming, so for the purposes of efficiency, experienced users tend to create new types and add multiple terms in batches. Figure 5.29 shows that by adding the concept **CDs**, there are now 140 instances of concepts in 115 records that were matched with the **music** type.

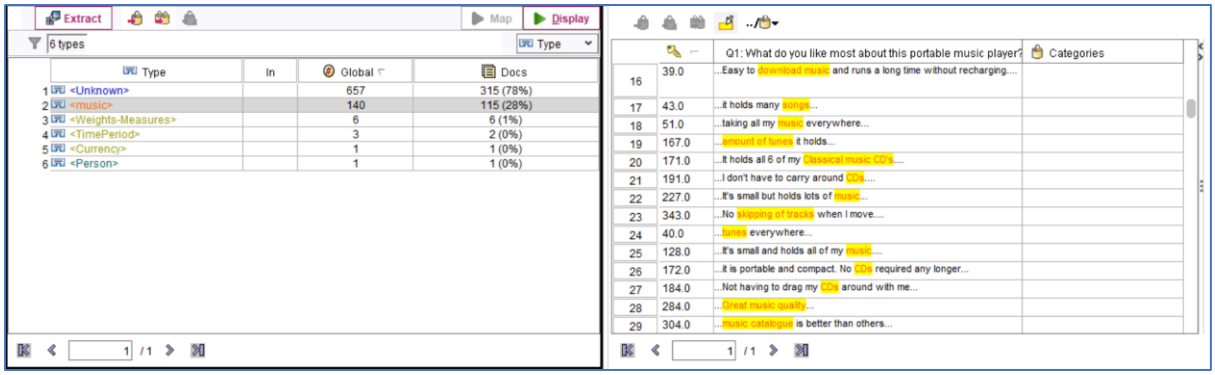


Figure 5.29 The updated extraction pane type list showing 140 concepts in 115 records in the 'music' type

Although, the business of creating, editing and adding new concepts to types can be one of the most protracted aspects of working with Modeler Text Analytics, it is arguably the single most valuable use of the analyst's time, as creating a comprehensive resource of identified concepts is one of the most effective ways to categorise and analyse text data. Furthermore, this is a highly iterative process whereby the tool incrementally captures and categorises more and more terms and phrase, so that over time, the application itself becomes increasingly accurate and more useful.

5.3 Forcing extraction with new types

As we discussed earlier, Model Text Analytics employs a parts of speech algorithm to identify candidate terms for extraction. Moreover, this system is deliberately weighted towards the extraction of terms that contain nouns. As a consequence, when reviewing the source text, analysts can clearly see phrases and words that the extractor has ignored. To do this, within the category pane in the top left-hand side of the interactive workbench:

Highlight the root category marked All Documents

And click the button marked:

The data pane now displays all the individual text responses to the survey. Note that the terms that have been extracted are colour coded according to their type group (see Figure 5.30).

		Q1: What do you like most about this portable music player?
175	69.0	...Stores 5000 songs in a compact portable player....
176	73.0	...its design is way cool...
177	306.0	...Quite stylish and easy to use....
178	144.0	...The size makes it convenient to carry around....
179	310.0	...the colour of the device...
180	148.0	...The online store is great. Also, sound quality is excellent....
181	298.0	...Uses standard batteries...
182	152.0	its small
183	302.0	...Great sound....
184	156.0	you can take it everywhere
185	128.0	...It's small and holds all of my music....
186	294.0	...Ease of use...
187	132.0	...size...
188	136.0	...The amount of music files it holds...
	140.0	...Controls are easy to understand and use. It's a well-designed,

Figure 5.30 Text responses displayed in data pane

You can also see that many terms, especially those which are simple adjectives, have not been extracted. For this reason, phrases such as **small** or **easy** have been ignored whereas phrases like **ease of use** or (the) **size** have been extracted. One way to force the extractor to extract these terms, is to include them in a type grouping.

To illustrate this, we can create three new types for the local library based on *small size*, *ease of use* and *portability*. Figure 5.31 shows the type properties associated with each of these new types.

Name	Match	Inflected	Colour	Terms
small	Entire and Any	Yes	Green	small, tiny, its little, little size, compact
easy	Entire and Any	Yes	Purple	ease of use, easy, user friendly, user friendliness, intuitive
portable	Entire and Any	Yes	Light Blue	portable, mobility, fits in my pocket, carry around, take it anywhere, take it everywhere

Figure 5.31 Properties of three new types

Figure 5.32 shows the resources pane for the local library with these three new types added.

Resources

Term	Match	Inflect	Type	Library
portable	Entire and Any	☑	portable	Local Library
mobility	Entire and Any	☑	portable	Local Library
fits in my pocket	Entire and Any	☑	portable	Local Library
carry around	Entire and Any	☑	portable	Local Library
take it anywhere	Entire and Any	☑	portable	Local Library
take it everywhere	Entire and Any	☑	portable	Local Library
ease of use	Entire and Any	☑	easy	Local Library
easy	Entire and Any	☑	easy	Local Library
user friendly	Entire and Any	☑	easy	Local Library
user friendliness	Entire and Any	☑	easy	Local Library
intuitive	Entire and Any	☑	easy	Local Library
small	Entire and Any	☑	small	Local Library
tiny	Entire and Any	☑	small	Local Library
its little	Entire and Any	☑	small	Local Library
little size	Entire and Any	☑	small	Local Library
compact	Entire and Any	☑	small	Local Library
music	Entire and Any	☑	music	Local Library
tune	Entire and Any	☑	music	Local Library
song	Entire and Any	☑	music	Local Library
track	Entire and Any	☑	music	Local Library
cds	Entire and Any	☑	music	Local Library

Figure 5.32 Resources window for the Local library with now containing four type groupings

If we now return to the categories and concepts window, we will need to re-run the extractor to take account of the changes made to the current resource template. Figure 5.33 shows the type view in the extraction pane, which now indicates that the three new types matched with several responses to the extent that **small** captured 69 records, **easy** 66 records and **portable** 27 records.

Extract | Map | Display

9 types | Type

	Type	In	Global	Docs
1	<Unknown>		618	301 (74%)
2	<music>		138	113 (28%)
3	<small>		69	69 (17%)
4	<easy>		68	66 (16%)
5	<portable>		28	27 (7%)
6	<Weights-Measures>		6	6 (1%)
7	<TimePeriod>		3	2 (0%)
8	<Currency>		1	1 (0%)
9	<Person>		1	1 (0%)

Figure 5.33 Type view in extraction pane showing the match frequency of responses and records of the three new type groups

5.4 Closing the interactive workbench

At this point, the analyst may be wondering how they may save their progress so that they can return to their work later. In fact, there a number of options available. Later

we will look at how we can save and share individual libraries, resource template and entire projects, but for now, let's turn our attention to what happens when we choose to close the interactive workbench.

What is important to understand, is that although the workbench appears as an IBM SPSS Modeler output object, in reality, it is a special environment that in many ways acts as an application in and of itself. For this reason, analysts should be careful not to have too many separate instances of interactive workbenches open, as Modeler's memory allocation can become rapidly depleted.

To close the interactive workbench, simply click:

File

Close

We are now presented with a dialog offering three separate options: **Update**, **Exit** or **Close** (see Figure 5.34). Let's address the last two options first.

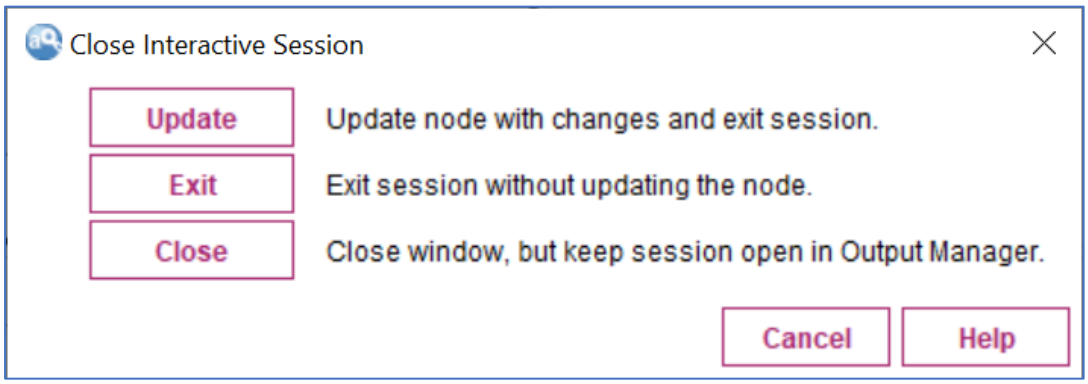


Figure 5.34 The Close Interactive Session dialog offering three alternative close options

Exit: This option will completely close the session. Any work including extraction results, alterations to the resource template or created categories are discarded and the interactive session is deleted from the output tab. ***This option should therefore be avoided if the user wishes to continue working where they left off but have not yet saved any of their edited resource files separately.***

Close: This option will close the interactive session window but the session itself will remain open in the background. Users can simply click on it in the output tab to re-join it later***. Again, care should be taken with this option as a subsequent system crash can lead to any unsaved work being lost.***

Update: Choosing this special option actually generates other choices for the user. The update option allows users to *cache their work* in the relevant session's

associated *text mining node*. Choosing this option however, also results in the session window being closed and the interactive workbench session being deleted form the output tab.

Figure 3.35 shows an annotated image of the sub-dialog (labelled **Save and Exit**) that is generated when the **Update** option is clicked.

Figure 5.35 The Save and Exit sub-dialog that appears when the Update option is clicked when closing an interactive session

By clicking **Update** on closing the session, we have a few additional options that we may choose from:

- **Keep only session work (categories, patters, resources, etc.):** This option keeps any of the additions or edits you made to the session's resource files. This would include new types, categories, changes to synonyms or the advanced resource settings. The alterations will be added to the text mining node so that you can continue the session at a later time. What is not saved using this option, are the actual extraction results themselves. Re-initiating the session having chosen this option will require the extraction process to be re-executed (although any changes made to the resources will take effect, so the user won't be starting from scratch).
- **Keep session work and cache text data with extraction results for reuse**: This option works in the same way as the previous, except that with this method, the text data itself, along with the extracted results *are also cached* in the text mining node. Using this option offers a quick way to return to a previous interactive session without having to re-extract the data.
- **Publish libraries**: Later in the course we will explore the concept of publishing in more detail. For now, suffice to say that publishing enables the user to make the libraries from this session available to other sessions and projects dealing with similar data.

Users should also know that both of these update options are available from *within* an interactive workbench session by choosing **File > Update Modeling Node** in the main menu. Lastly, we should bear in mind that if we choose *either* update option, it will of course be necessary to *save the modeler stream* containing the node before exiting IBM SPSS Modeler entirely.

To demonstrate this functionality, within the interactive workbench click:

File

Close

Update

From the **Save and Exit** dialog, click the second option (as shown in Figure 5.36):

Keep session work and cache text data with extraction results for reuse

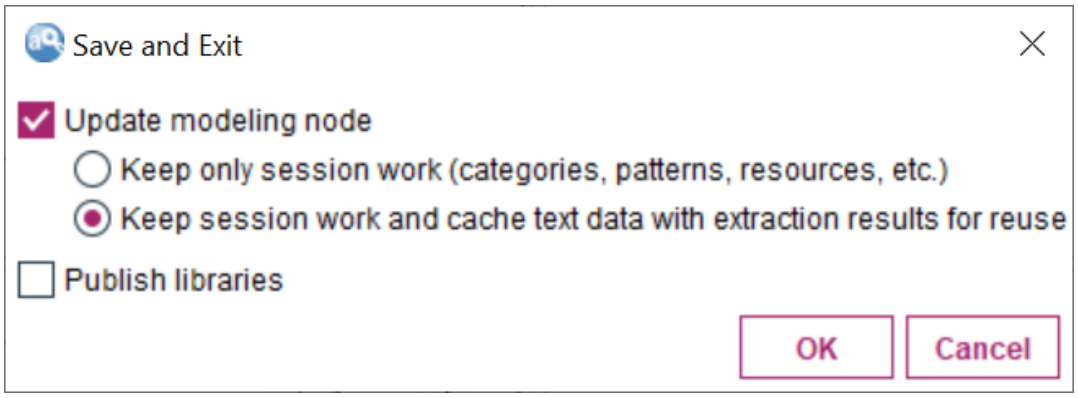


Figure 5.36 The Save and Exit sub-dialog having clicked the Update option – text data and extraction results will be saved as well as session work

A message dialog appears informing us that the modelling node has been updated as shown in Figure 5.37.

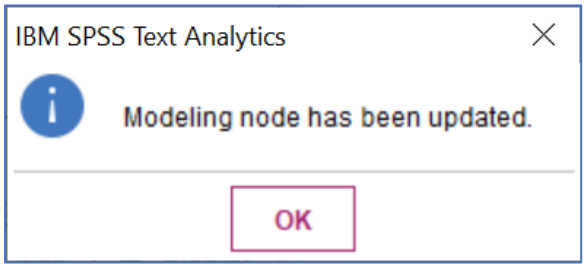


Figure 5.37 Message informing us the modelling node has been updated

The interactive workbench session closes. If we now right-click to edit the text mining node and browse the model tab, as Figure 5.38 shows, we can see that two check box options have been pre-filled.

The first check-box option indicates that the interactive workbench will use any session work that was saved from the previous update. The second option indicates that no extraction process should be run and that the session should re-use the cached data and extraction results from the previous update.

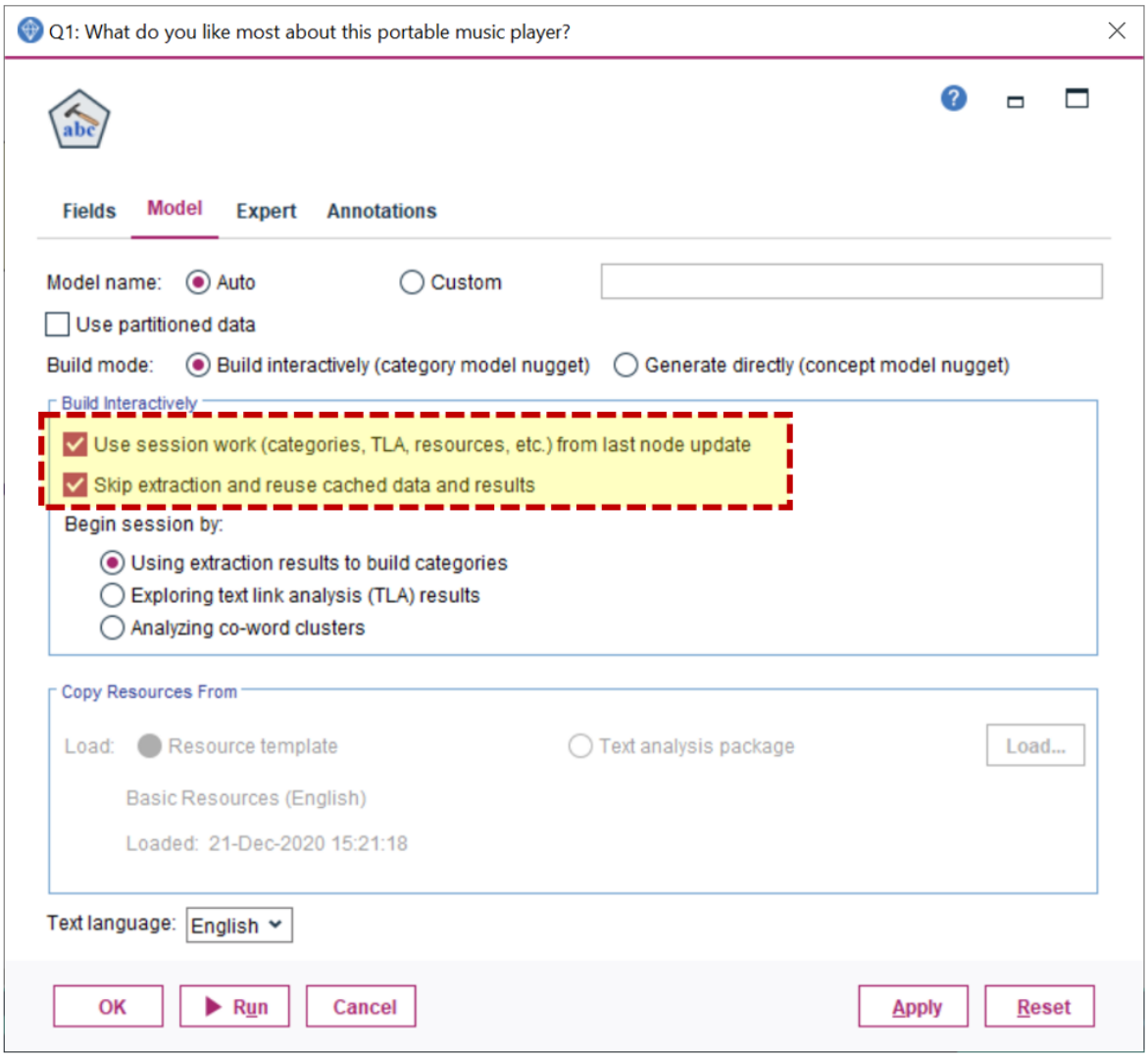


Figure 5.38 The Text Mining node

To show that we can restore the data and session work from the update when we previously closed the workbench, click:

Run

Now an information dialog appears (see Figure 5.39) telling us that the session will be restored using the cached data and extraction results from the previous. It warns us that if, in the meantime, we have connected a new data source or added additional data that this new information will be ignored.

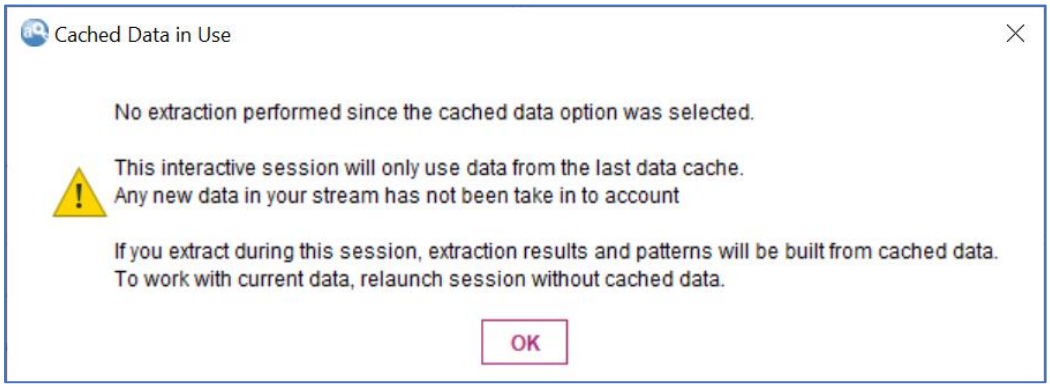


Figure 5.39 Cached Data in Use dialog – the session will be restored using the data, extraction results and session work from the previous interactive workbench session

To continue, click:

OK

Figure 5.40 shows the interaction pane from this fresh interactive session displaying the extracted results and type groups that were saved in the updated node when the previous session was closed.

	Concept	In	Global	Docs	Type
1	music		54	52 (13%)	<music>
2	easy		50	49 (12%)	<easy>
3	small		45	45 (11%)	<small>
4	use		38	37 (9%)	<Unknown>
5	song		30	26 (6%)	<music>
6	size		26	26 (6%)	<Unknown>
7	product		18	17 (4%)	<Unknown>
8	ease of use		15	15 (4%)	<easy>
9	cds		13	13 (3%)	<music>
10	portable		12	12 (3%)	<portable>
11	sound quality		11	11 (3%)	<Unknown>
12	store		11	11 (3%)	<Unknown>
13	light		11	11 (3%)	<Unknown>
14	portability		10	10 (2%)	<Unknown>
15	ability		8	7 (2%)	<Unknown>
16	design		8	8 (2%)	<Unknown>
17	small size		8	8 (2%)	<small>

Figure 5.40 Restored data, extraction results and session work following the closure of the previous workbench session

Practice Exercise – Chapter 5

Within the folder **Student Exercises** open the following stream:

Chapter_05_Practice.str

1. Run the text mining node connected to the data source reading the file **Rental Car Satisfaction**. As this node is using the **Basic Resources** template, once the extraction process is complete, we will see that most of the concepts are typed as **unknown**.

2. Switching to the **Resource Editor** window, define the following new types in the local library.

Type Name	Match	Inflected	Colour	Terms
air travel	Entire and Any	Yes	Blue	airport, plane, terminal, airline
location	Entire (no compounds)	Yes	Purple	location, proximity, closeness, close to
mileage	Entire and Any	Yes	Red	mileage, miles

3. Switching back to the **Categories and Concepts** window, re-extract the concepts and check that the new type groups are working correctly. Do you notice any mistakes?

4. Within the extraction pane, sort the concepts in alphabetical order and find the concept **fly**. Right-click on the concept and add it to the appropriate type group.

5. Define *at least two more type groups of your own choosing*. Suggested topics include financial factors, availability or convenience.

6. When you are finished, close the session but choose: **Update** then choose, **Keep only session work (categories, patterns, resources, etc.)** and click **OK.**

7. Overwrite the existing stream by saving it.

Chapter 6 Working with resource files

So far, the examples that we've looked at have been based on the Basic Resources (English) template. In this chapter, we can see the effects of using a pre-built resource template that already contains a number type dictionaries. Figure 6.1 shows a table listing the various English language resource templates that ship with IBM SPSS Modeler Premium Version 18.2.1.

Template	Notes
Basic Resources	The most basic English language template
Opinions	The Opinions template builds on the Basic Resources template to include a number of libraries designed to capture expressions of positive and negative sentiment
CRM	A template containing a CRM (Customer Relationship Management) library with several type groups related to sentiment, customer terminology, procedures, resolutions, communication etc
Ads Opinions	A version of the Opinions template that includes a library devoted to concepts related to advertising
Bank CRM	A version of the CRM template with a banking library containing type groups related to concepts like transactions, card payments and personnel
Bank Satisfaction Opinions	A version of the Opinions template that includes libraries related to finance, information and customer satisfaction
Bioscience	A version of the Basic Resources template augmented with three libraries related to medicine, chemicals and genetic proteins
Customer Satisfaction Opinions	A version of the Opinions template that includes libraries related to customer satisfaction, product satisfaction and communication
Demographics	A version of the Basic Resources template augmented with two libraries containing type groups related to demographics and industry sectors
Employee Satisfaction Opinions	A version of the Opinions template that includes a library containing concepts related to employee satisfaction
Gene Ontology	A version of the Basic Resources template with a library a type group containing over four thousand terms related to genes
Genomics	A version of the Basic Resources template with a library containing over thirty thousand terms related to genomics

Hotel Satisfaction	A version of the Opinions template including two libraries containing type groups related to the hotel industry and food
Insurance CRM	A version of the CRM template including a library containing type groups related to the insurance industry
IT	A version of the Basic Resources template with a library containing over a thousand terms related to computing and information technology
Market Intelligence	A version of the Basic Resources template including a library containing several type groups with terms related to commercial enterprises, stock exchanges, executive functions and financial activity
MeSH	Another life sciences resource based on the Basic Resources template but including a library containing over ninety thousand terms from the Medical Subject Headings (MeSH) thesaurus
Patient Satisfaction	A version of the Opinions template with an additional library containing type groups related to physicians, medical procedures, amenities and nursing staff
Product Satisfaction Opinions	A version of the Opinions template with two additional libraries built on the themes of information and products and containing type groups related to documentation, product features, performance and characteristics
Security Intelligence	A version of the Basic Resources template including a library of type groups covering topics such as weaponry, crimes, transportation, airlines and over two thousand relevant organisations
TelCo Satisfaction	A version of the Opinions template including a telecommunications library of type groups covering subjects such as phones, service plans, reception issues and accessories
TroubleShooting CRM	A version of the Basic Resources template with a library related to conditions, actions and problems. This template also contains two libraries related to cars and automotive components.

Figure 6.1 List of pre-built resource templates that ship with IBM SPSS Modeler Premium

Resource templates are usually collections of multiple resource files including libraries containing various ‘dictionaries’ of type groups. These terms are also linked

with other dictionaries containing synonyms, excluded terms and optional elements (see Figure 6.2). Lastly, a resource template can make use of advanced resources that control things like extraction patterns, non-linguistic entities and abbreviations.

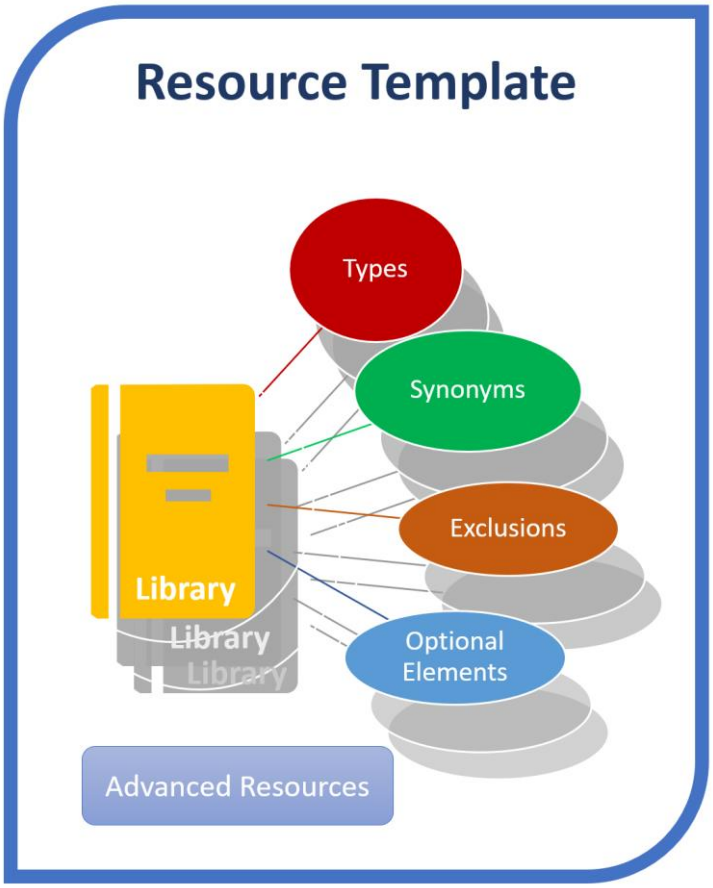


Figure 6.2 Resource Templates contain collections of advanced resources and libraries which in turn link to various dictionaries of types, exclusions, synonyms and optional elements

6.1 Loading a resource template

Using the stream **06_Working_With_Resources.str** we can illustrate the effect of using a resource template on the extraction process. To load a new resource template for extraction:

Right-click on the text mining node

Browse the Model tab

And click:

Load

The **Load Resource Template** dialog is displayed showing all the currently available resource templates. From the list of resource templates select:

Opinions (English)

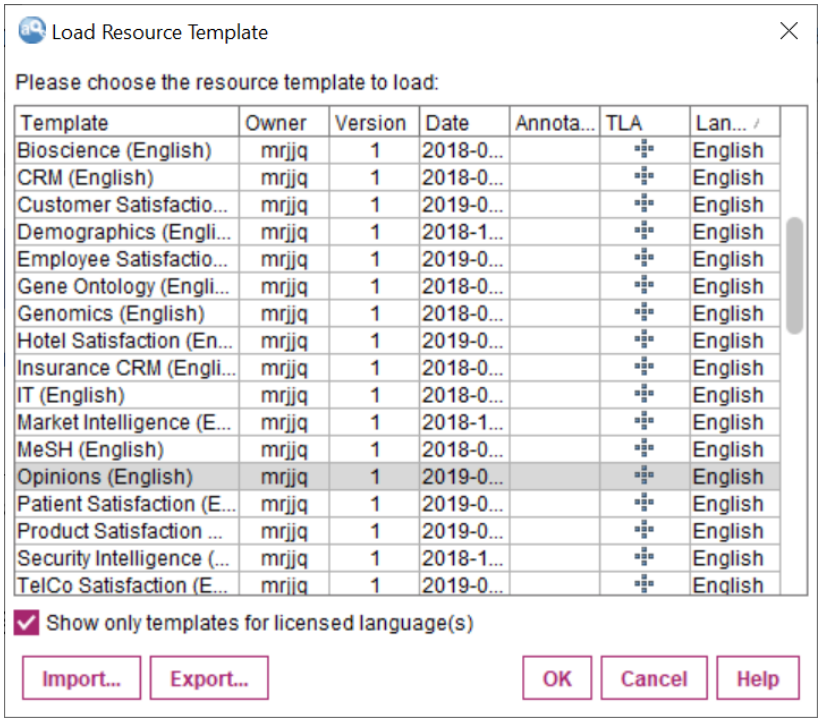


Figure 6.3 Load Resource Template dialog

To load the Opinions resource template, click:

OK

Now click:

Run

When the extraction is complete, we can immediately see from the colour-coded concepts in the extraction pane, that the Opinions template has identified a number of terms as belonging to the positive type group.

	Concept	In	Global	Docs	Type
1	small		58	58 (14%)	<Contextual>
2	like		55	43 (11%)	<Positive>
3	music		53	51 (13%)	<Unknown>
4	easy to use		45	44 (11%)	<Positive>
5	portable		43	42 (10%)	<Positive>
6	excellent		40	33 (8%)	<Positive>
7	size		36	36 (9%)	<Unknown>
8	good		30	29 (7%)	<Positive>
9	listening		30	29 (7%)	<Unknown>
0	songs		29	26 (6%)	<Unknown>
1	sound quality		21	21 (5%)	<Unknown>
2	product a		19	18 (4%)	<NewType>
3	large		16	16 (4%)	<Positive>
4	design		16	16 (4%)	<Unknown>
5	cds		12	12 (3%)	<Unknown>
6	lightweight		12	12 (3%)	<PositiveFeeling>
7	light		12	12 (3%)	<Positive>

Figure 6.4 Extraction pane showing terms identified as positive due to use of Opinions template

To view the contents of the Opinions resource template, switch to the Resource editor window by selecting it from the drop-down menu:

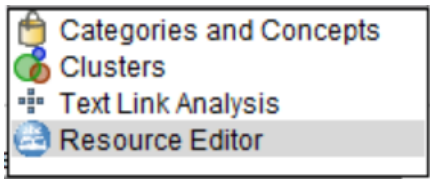


Figure 6.5 shows an annotated view of the Resource editor window.

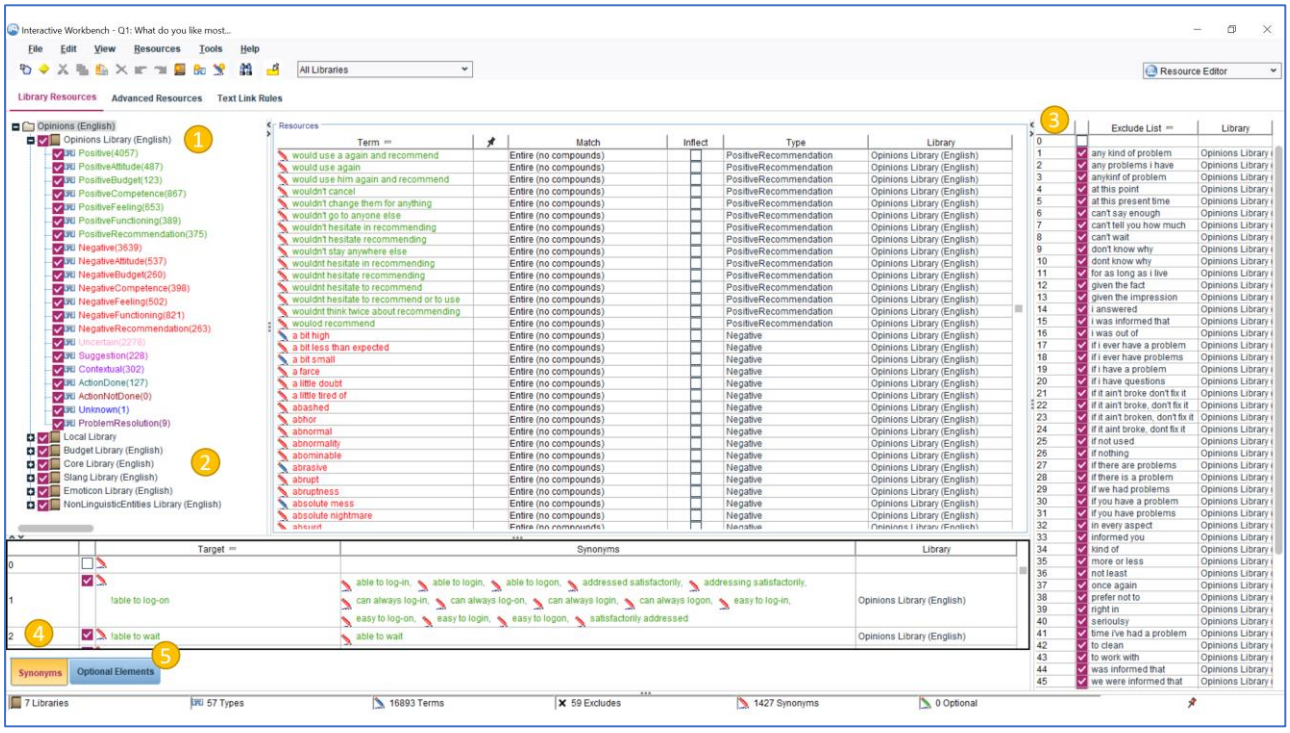


Figure 6.5 The Resource editor window

The Resource editor window contains a number of panes that allow the user to edit the contents of the various libraries and dictionaries in a given resource template.

1. **Opinions Library**: This is the main Opinions library within the Opinions resource template. It contains thousands of terms split over twenty type groups mostly related to positive and negative sentiments as well as some contextual type groups.
2. **Other Libraries**: These are the other libraries in the resource template, including the compiled dictionaries in the core library. Note that this includes a library for slang terms, emoticons, budgetary (financial) terms and non-linguistic entities like phone numbers and timestamps.
3. **Exclude List**: There may well be re-occurring phrases or terms that the analyst is not interested in extracting. They might relate to page numbers, email signatures, slogans or errors. The exclude list is a dictionary of terms that we can edit to ensure that these kinds of concepts are ignored.
4. **Synonyms**: The synonym dictionary (part of the substitution dictionaries in a resource template) consists of two elements: a target term and a list of synonyms. When the system encounters a term that occurs in the synonym list, it replaces it with the appropriate target term. Sometimes the synonyms are used to deal with frequently misspelled words, for example as Figure 6.6 shows, the target term **friendly** is used to replace several misspelled versions of this word. In a similar vein, terms such as **superfriendly** and **non-hostile** are substituted for this target term. You may also encounter situations where you don't want a term to be treated as if it was a synonym of a target term. If for example, you feel that the word **sympathetic** is not a legitimate synonym of **friendly**, you could simply delete it from the list of synonym terms.
5. **Optional Elements**: Rather link synonyms, optional elements help to group similar terms under one target concept. These are also part of the system's substitution dictionaries. Optional elements work by identifying words in compound terms that should be ignored during extraction so that similar terms are identified as referring to the same thing. If for example you wanted the terms **microsoft** and **microsoft corp** to be treated the same way, you could simply declare the term **corp** to be an optional element.

Figure 6.6 The synonym dictionary showing synonyms of the target term 'friendly'.

With regard to substitution dictionaries, you should bear in mind that you can always add spelling variants, synonyms and compound variations to a type group and they will typed in the same way. So it's not always essential that you make use of synonyms and optional elements. However, if you find that term is not being correctly typed, it could be because it is already defined as a synonym of another term and is therefore being substituted during the extraction process.

The resource editor has its own resource menu where users can switch to a new resource template, add new libraries and generally manage the available resource files. To see the effect of switching to a new resource template, from the main menu, click:

Resources

Switch Resource Templates

An information dialog appears, warning us that the current resources will be cleared, so we may wish to save them if any changes have been made.

Figure 6.7 Warning information that the current resources are about to be replaced

To continue, click:

OK

From the **Switch Resources** dialog, choose:

Product Satisfaction Opinions (English)

And click:

OK

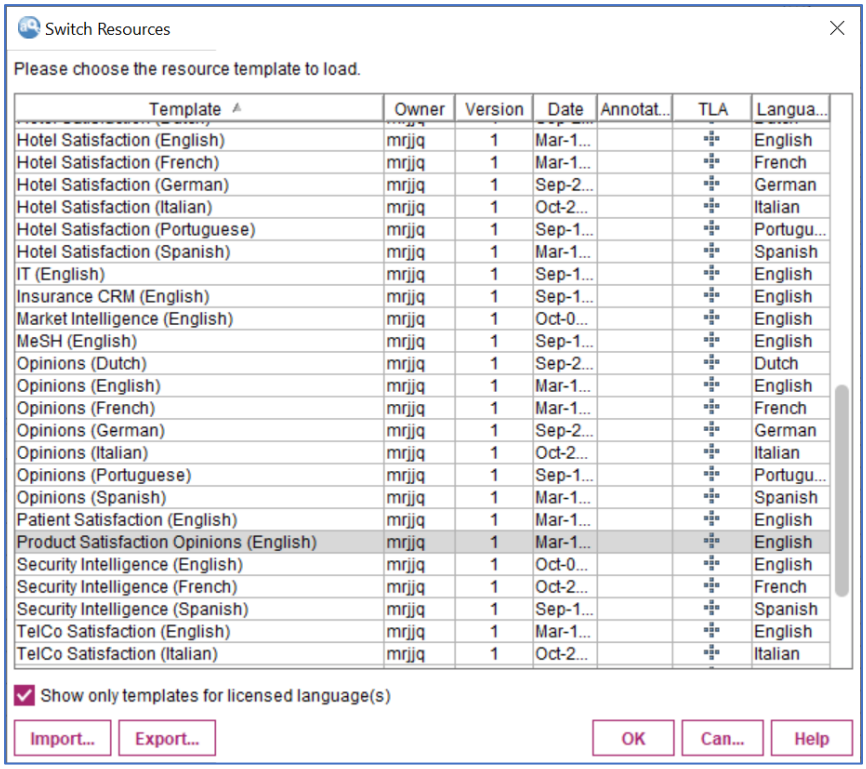


Figure 6.8 Switching to the Product Satisfaction Opinions (English) resource template

6.2 Synchronising resources

Before the resource template is switched, it's possible that another dialog will appear offering us the opportunity to *update* a library (see Figure 6.9). Later, we will take a closer look at how we can manage and share resources such as libraries. For now though, it's useful to understand that whenever a new interactive session is launched, a *copy* of that session's resource template is loaded for use. This means that when we make changes to the resources during an interactive session, we are really making changes to the currently loaded copy. The issue here is that we can end up with two copies that are out of sync with one another i.e., the current live **local** copy, and the original **public** copy.

Obviously, when we first install the software, all the resources are effectively *public* resources. But as we know, when we continue to work within various interactive sessions, changes can be made to libraries and dictionary content. Moreover, if we wish to make a recently edited resource publicly available to all future sessions, we need to **publish** it. If we don't publish a resource, the software will alert us to situations when the specific resource files are out of synch. There are situations when the local version is more recent that the public version (as denoted by an icon of an upward arrow). However, because it's possible to have two interactive sessions open at once, if you publish any changes in one session, you can also end up with a situation where the *public version* is more recent than the local instance currently

open in the *other* interactive session. As Figure 6.9 shows, this particular situation is denoted by a downward arrow icon next to the Opinions library, where the dialog is offering us the opportunity to update the local Opinions library with the more recent public version so that they are in sync before we switch resources.

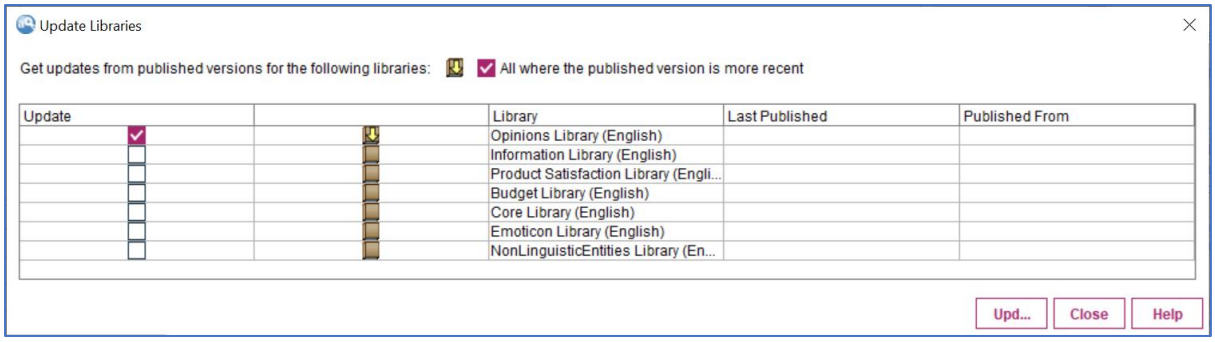


Figure 6.9 Update libraries dialog showing an instance where a public library is more recent than a local one

Figure 6.10 shows the various icons that are associated with each permutation of synchronous and asynchronous resources.

Icon	Status	Description
	**Unpublished**	The local library has never been published
	**Synchronized**	The local and public libraries are identical
	**Out of date**	The public version is more recent than the local version
	**Newer**	The local version is more recent than the public version
	**Out of sync**	The local and public libraries both contain changes that the other does not. The user must decide whether to update the local instance of the library with the public version, or to publish the local instance and overwrite any changes in the currently public version

Figure 6.10 Library synchronisation states

In this case, we don't need to make any changes to this resource template, so we can click:

Close

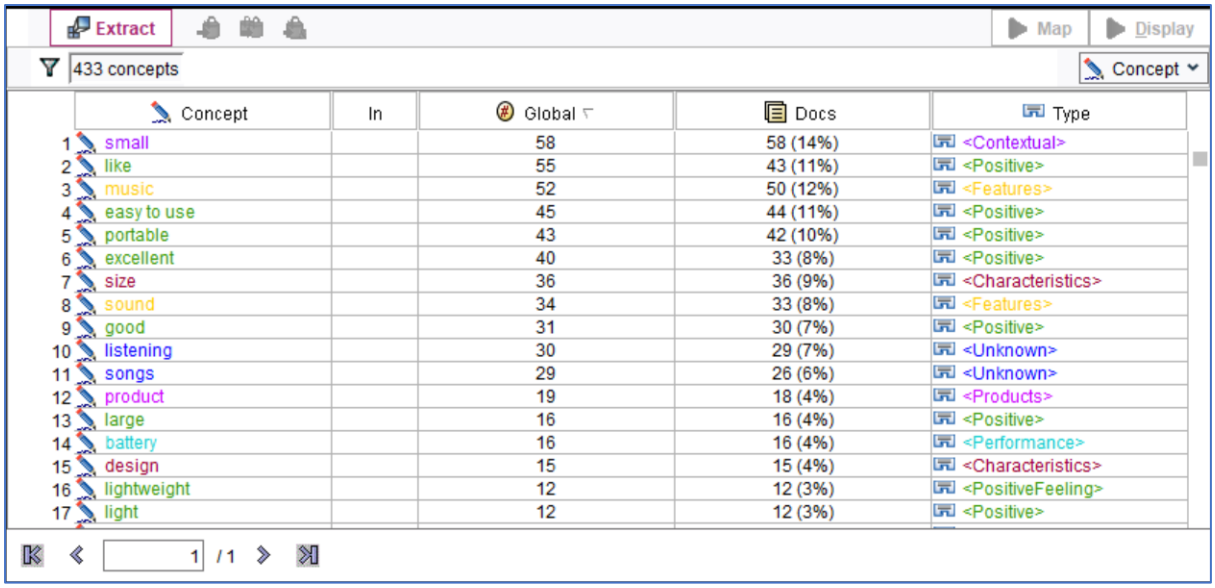
6.3 The Product satisfaction template

We can now see that resource template has been switched, as **Product Satisfaction** has been added to the collection of libraries. Returning to the categories and

concepts window, we can also see that the extracted concepts and data have disappeared. We will therefore need to re-extract the concepts so that the new resource template can take effect. To do so, click:

Extract

As Figure 6.11 shows, the extracted results now contain a number of new type groups such as **features**, **characteristics** and **performance**.



	Concept	In	Global	Docs	Type
1	small		58	58 (14%)	<Contextual>
2	like		55	43 (11%)	<Positive>
3	music		52	50 (12%)	<Features>
4	easy to use		45	44 (11%)	<Positive>
5	portable		43	42 (10%)	<Positive>
6	excellent		40	33 (8%)	<Positive>
7	size		36	36 (9%)	<Characteristics>
8	sound		34	33 (8%)	<Features>
9	good		31	30 (7%)	<Positive>
10	listening		30	29 (7%)	<Unknown>
11	songs		29	26 (6%)	<Unknown>
12	product		19	18 (4%)	<Products>
13	large		16	16 (4%)	<Positive>
14	battery		16	16 (4%)	<Performance>
15	design		15	15 (4%)	<Characteristics>
16	lightweight		12	12 (3%)	<PositiveFeeling>
17	light		12	12 (3%)	<Positive>

Figure 6.11 Re-extracted concepts using the Product Satisfaction Opinions template

Clearly, using an appropriate pre-built template can help save a lot of time, simply because these resources may already contain many typed concepts that we would otherwise have needed to identify manually with the interactive workbench. Even if the types themselves appear at first a bit generic, we can still use the concepts to create new more specific types. Also note the effect of the synonyms in the substitution dictionary by hovering the cursor over an existing extracted concept. To illustrate this:

***Move the cursor to the concept* portable**

Figure 6.12 shows an accompanying pop-up label which indicates that this concept also contains the underlying terms **can take it anywhere**, **can take it everywhere**, **easy to bring along** and **easy to carry**.

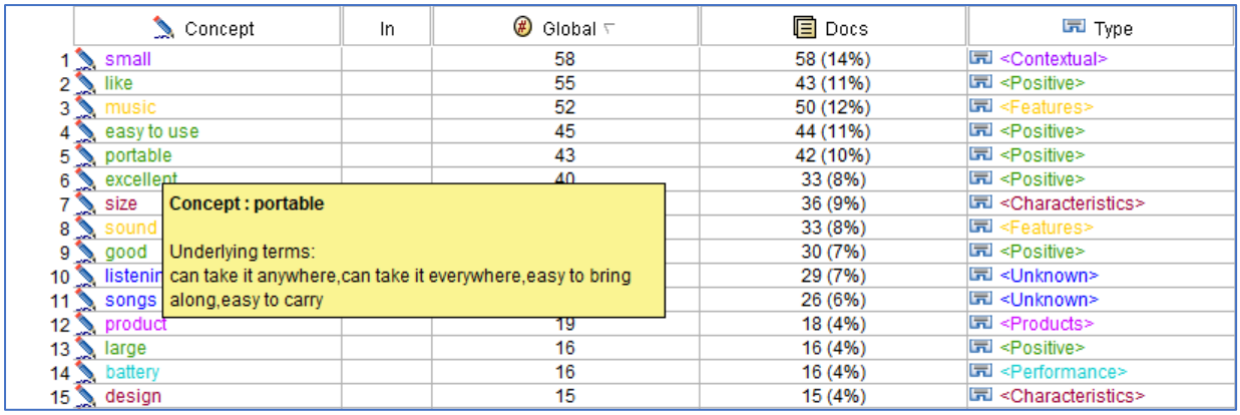


Figure 6.12 Underlying terms provided by the synonym dictionary associated with the concept 'portable'

Nevertheless, you may not always agree with the way in which concepts have been typed or the synonyms attached to them. If we hover the cursor over the term **easy to use**, we can see that one of the underlying terms is **can be taken anywhere** (see Figure 6.13). Surely this should be a synonym of portable?

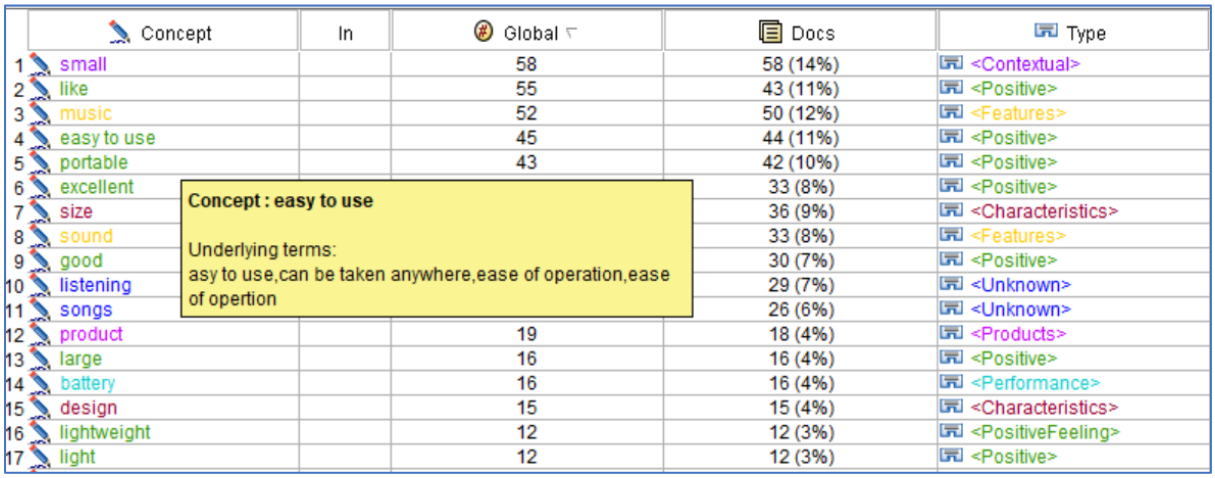
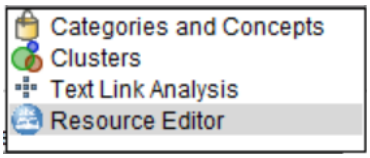


Figure 6.13 Underlying terms provided by the synonym dictionary associated with the concept 'easy to use'

Of course, the phrase **can be taken anywhere** may not occur in the data itself, but we can still correct it in the resource template for future analyses. To do this, we need to return to the resource editor by clicking:



Within the resource editor, from the main menu, click:

View

Find

In the opened search text box enter the phrase:

can be taken anywhere

Click the adjacent button:

Find

The first occurrence of the phrase in found in the positive type group. To find the second occurrence, once again click:

Find

Now the phrase is found and highlighted in the synonym dictionary, where it will be substituted by the target concept **ease of use**. To remove the term:

Double-click on the phrase

You will see a pop-up text window appear (see Figure 6.14), to remove the phrase simply highlight it and cut it by using the keyboard short-cut

Ctrl-X

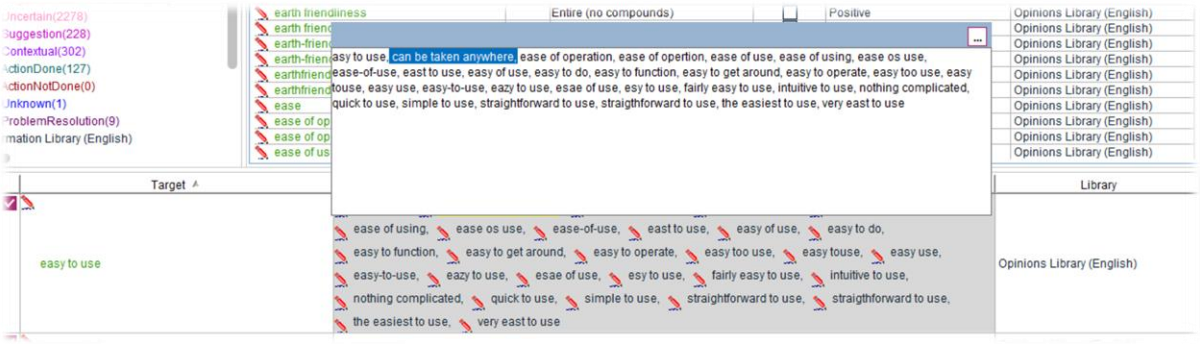


Figure 6.14 The phrase 'can be taken anywhere' highlighted in the synonym dictionary where it will be substituted by the term 'easy to use'

To close the box hit:

Enter

Return to the search box and type:

Portable

Repeat the search steps and open the window containing the list of synonyms associated with the term **portable**.

Paste the cut compound term (separated by comma after the last phrase) using the keyboard short-cut:

Ctrl-V

Figure 6.15 shows this completed step.

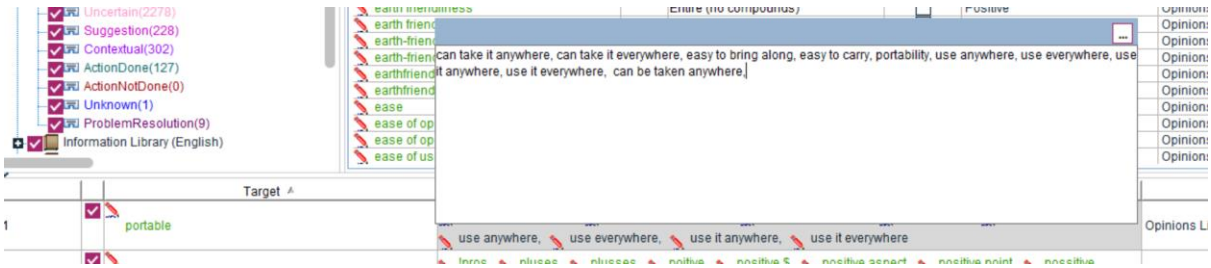


Figure 6.15 The phrase 'can be taken anywhere' pasted into in the synonym dictionary where it will be substituted by the term 'portable'

To continue, hit:

Enter

Having changed the resource template, the local version is now more up to date than the public one. To reconcile this:

Right-click on the Opinions library as shown in Figure 6.16 and click:

Publish Libraries

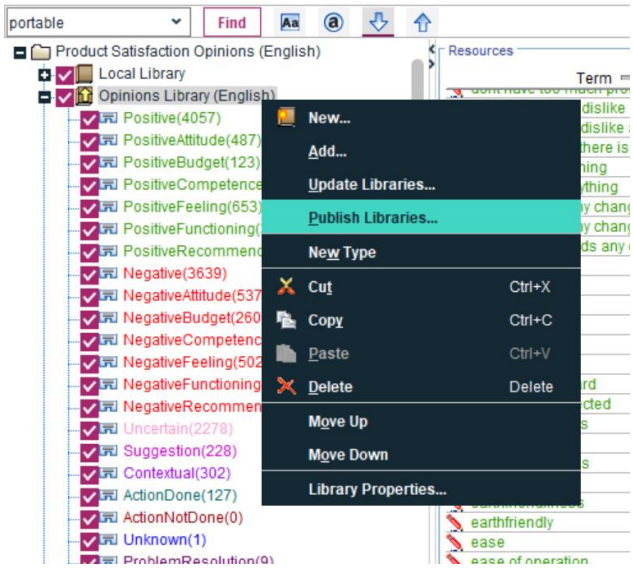


Figure 6.16 Selecting the recently edited local version of the Opinions library for publishing

From the pop-up dialog shown in Figure 6.17 click:

Publish

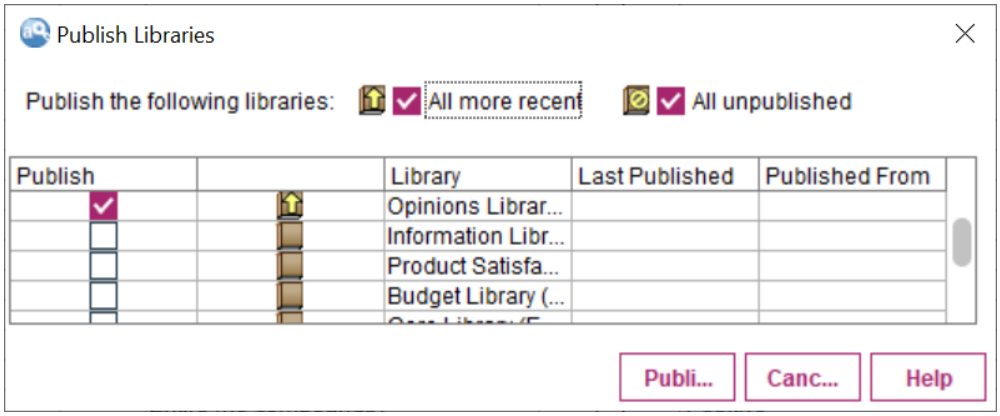


Figure 6.17 Publishing the recently edited local version of the Opinions

After the local version of the library has been published, we can return to the categories and concepts window where the extraction pane is again shaded in yellow indicating that the extractor will need to be run again. In which case, click:

Extract

Despite the occasional inconsistency like the one we just encountered, using the Product Satisfaction Opinions resource template has resulted in a lot of useful concepts being extracted. We can see that many concepts have been typed using generic type groups such as **positive**, including concepts like **easy to use**, **light**, **compact** and **cool**. We should remember that these are responses to a question which asked people to state what they like *most* about a product, so we would expect a lot of positive terms.

We can also see that there are still a number of concepts belonging to the **unknown** type group. Even unknown concepts however, can make use of inflections and synonyms. For example, the unknown concept **friend** is linked to the following underlying terms **friends**, **mate** and **mates**. To focus in on this type group, we can launch the Extraction Filter dialog. Click:

Within the dialog, click the radio button:

Selected Types

From the list of types, choose:

<Unknown>

Figure 6.18 shows the completed dialog at this stage.

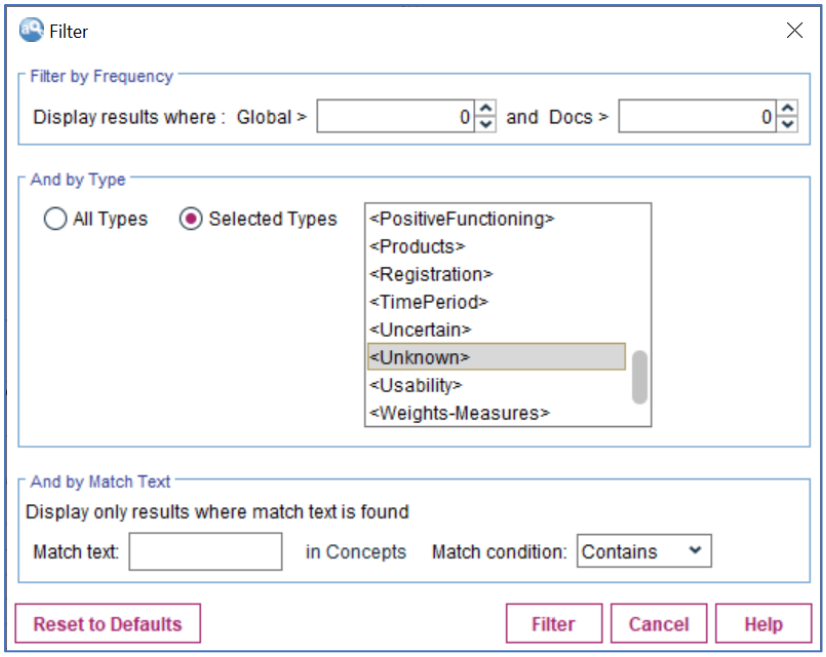


Figure 6.18 Filtering extracted concepts to show only those in the 'unknown' type group

To continue, click:

Filter

As Figure 6.19 shows there are still several useful concepts that could be used as the basis of a new type group or added to an existing one.

	Concept	In	Global	Docs	Type (Selected)
1	listening		30	29 (7%)	<Unknown>
2	songs		29	26 (6%)	<Unknown>
3	train		4	4 (1%)	<Unknown>
4	working		4	4 (1%)	<Unknown>
5	friend		4	4 (1%)	<Unknown>
6	fact		4	4 (1%)	<Unknown>
7	load		3	2 (0%)	<Unknown>
8	traveling		3	3 (1%)	<Unknown>
9	stores		3	3 (1%)	<Unknown>
10	use		3	3 (1%)	<Unknown>
11	son		3	3 (1%)	<Unknown>
12	tunes		3	3 (1%)	<Unknown>
13	work		3	3 (1%)	<Unknown>
14	web		3	3 (1%)	<Unknown>
15	room		2	2 (0%)	<Unknown>
16	tracks		2	2 (0%)	<Unknown>
17	pocket		2	2 (0%)	<Unknown>

Figure 6.19 Filtered view of 'unknown' concepts in the extraction pane

However, one of the lessons that text mining analysts learn early on, is that not everything deserves to be typed or given its own category. The most common unknown concept is **listening**. Does this really deserve its own type? Possibly, if it is associated with concepts like **positive listening experience**, but otherwise it's just a description of the main purpose of an audio player. It's often normal practice for the

most frequent concepts to be ignored by the analyst. For example, with hotel satisfaction surveys, the most common concept tends to be **hotel** and with complaints data about car rentals, the most common concept is often **rental**.

Looking at the other concepts in this list, we see words like **pocket** which could be added to a type group such as **size** or **portability**. We also see words related to capacity like **stores** and **room**, as well as terms that could be used to create new type groups related to journeys like **train** or **traveling**. In all these cases, it helps to check the context that the concepts are being used in by clicking on the extracted concept and viewing the accompanying data. To illustrate this, within the extracted concepts pane:

Use Ctrl-click to select the concepts **train** ***and*** **traveling**

Now click:

Display

Figure 6.20 shows the seven records that contain these concepts.

		Q1: What do you like most about this portable music player? (7)
1	145.0	...Listening while traveling to drown out the noise of crying kids!...
2	376.0	...Provides entertainment whilst traveling to and from work....
3	157.0	...huge selection of music at my fingertips when traveling. like bringing your whole stereo with you...
4	189.0	... the train, it allows me to sit back and relax and transition to home life....
5	163.0	... everyone else out on the train....
6	319.0	...I finally caved in after seeing all the commericals and seeing people on the train with Product...
7	388.0	...It has a radio in it so I can listen to the radio in the morning on the train and to music...

Figure 6.20 Records showing the context within which the concepts 'traveling' and 'train' occur in the data

To switch off the existing filter, once again click:

All Types

Filter

We can also begin reassigning concepts to new types directly from the unfiltered extraction pane. Right click on the concept:

easy to use

We've already seen that this concept has a number of synonym terms associated with it. Currently it is assigned to the **positive** type group. To reassign it to a new type group, from the pop-up menu select:

Add to Type

New…

Figure 6.21 shows this procedure.

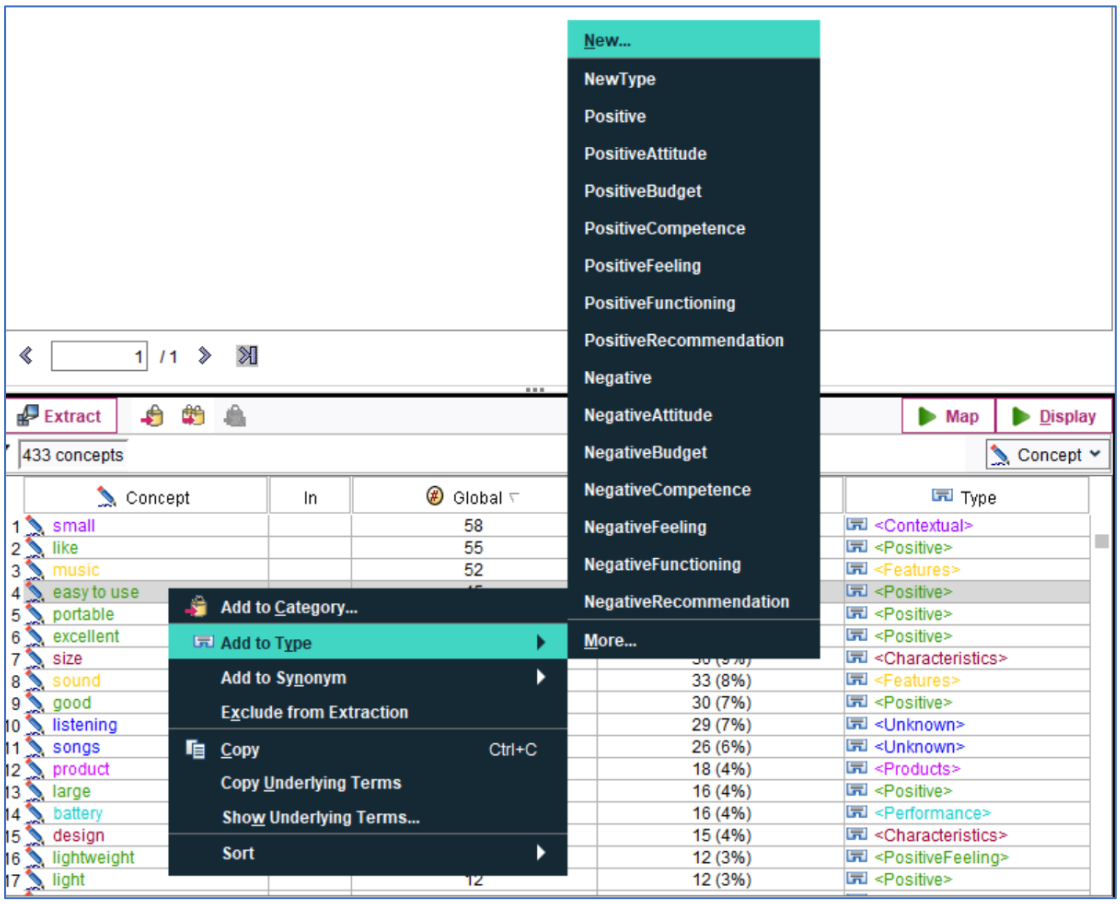


Figure 6.21 Adding the term 'easy to use' to a new type

The Type Properties dialog appears, and we can specify the new type group as shown in Figure 6.22.

Name	Match	Library	Inflected	Colour
Ease of use	Entire and Any	Local	Yes	Dark Green

Figure 6.22 Properties for new type group 'Ease of use'

Note that this is being assigned to our local library and that previously this term appeared in the Opinions library. To complete the specification, click:

OK

Immediately the **Resolve Conflicts** dialog appears informing us that this term already belongs to an existing type group (the **positive** type). This a common issue that users must address when they are assigning terms to type groups. Obviously, the term cannot be in both type groups, so we have to choose one over the other. Currently a black **x** icon appears next to the term, meaning that it will no longer be assigned to the **positive** type group (Figure 6.23).

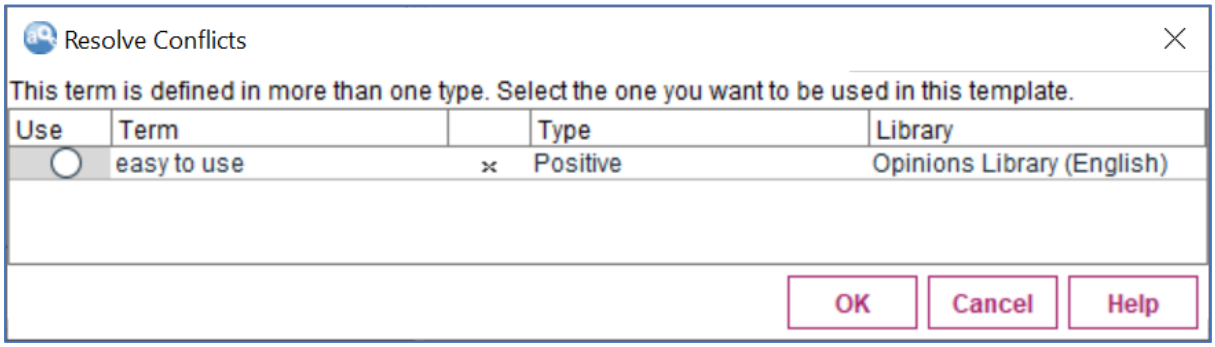


Figure 6.23 The Resolve Conflicts dialog with the forced term option selected

To continue simply click:

OK

The extraction pane is again shaded yellow to indicated that the extraction process would need to be re-run to take account of this change to the resources. If we briefly switch to the resource editor, we can see that the new type and term appear in the local library and that this red push-pin icon  is shown next to the term. This indicates that the term has been pinned (or *forced*) to this type group, which now takes precedent over the **positive** type group.

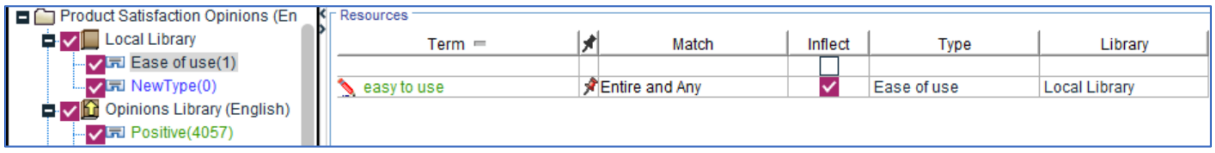
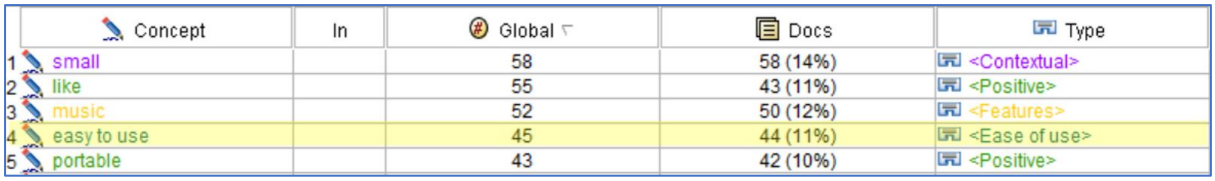


Figure 6.24 The term 'easy to use' forced to the type group 'Ease of use' as it appears in the resource editor

When we re-run the extractor, we can now see that the term is typed as **Ease of use** in the extraction pane (Figure 6.25).



	Concept	In	Global	Docs	Type
1	small		58	58 (14%)	<Contextual>
2	like		55	43 (11%)	<Positive>
3	music		52	50 (12%)	<Features>
4	easy to use		45	44 (11%)	<Ease of use>
5	portable		43	42 (10%)	<Positive>

Figure 6.25 The term 'easy to use' as it appears in the extraction pane

This approach of clicking on terms and reassigning them can be a productive way for analyst's to 'mop-up' concepts that have not been assigned yet or assigned correctly to type groups. We can repeat the process we just followed by re-creating a new type called **Portable** and assigning the term **portable** to it. Figure 6.26 shows the properties of this type group.

Name	Match	Library	Inflected	Colour
Portable	Entire and Any	Local	Yes	Light Blue

Figure 6.26 Properties for the type group 'Portable'

When we repeat this process however, something odd seems to have occurred. Firstly, when we select the concept **portable** in the extraction pane and request to view where it occurs in the data by clicking:

Display

We can see that when we move the mouse over the highlighted matching terms in the adjacent data pane, the pop-up label seem to show that many of the phrases are typed as **positive** rather than belonging to the **portable** type (see Figure 6.27). Rest assured that although compound terms like **can take it anywhere** are still present in the **positive** type group, they are replaced by their target synonym **portable** during extraction. So **portable** is the extracted *concept* and *it* is associated with the type group **Portable**.

		Q1: What do you like most about this portal	Categories
1	101.0	...It's portable....	
2	113.0	...It's portable...	
3	141.0	...portability...	
4	225.0	...that it is portable...	
5	385.0	...that it is portable...	
6	50.0	...portability...	
7	107.0	...It is portable....	
8	403.0	...can take it anywhere...	
9	156.0	...you can take it every	
10	316.0	...it portable...	
11	328.0	...portability...	
12	396.0	...Easy to carry...	
13	11.0	...It's portable! I can take it anywhere....	
14	353.0	...It's puny and easy to carry...	

Positive - portable

6.27 Terms associated with the concept portable as displayed in the

The second unusual result is illustrated by Figure 6.28 where we can see that the single concept **portable** now only occurs 36 times in 35 records when previously, as part of the **positive** type, it had occurred 43 times in 42 records.

	Concept	In	Global	Docs	Type
1	small		58	58 (14%)	<Context
2	like		55	43 (11%)	<Positive
3	music		52	50 (12%)	<Feature
4	easy to use		45	44 (11%)	<Ease
5	excellent		40	33 (8%)	<Positive
6	size		36	36 (9%)	<Charac
7	portable		36	35 (9%)	<Portabl

Figure 6.28 The concept 'portable' in the extraction pane after type reassignment

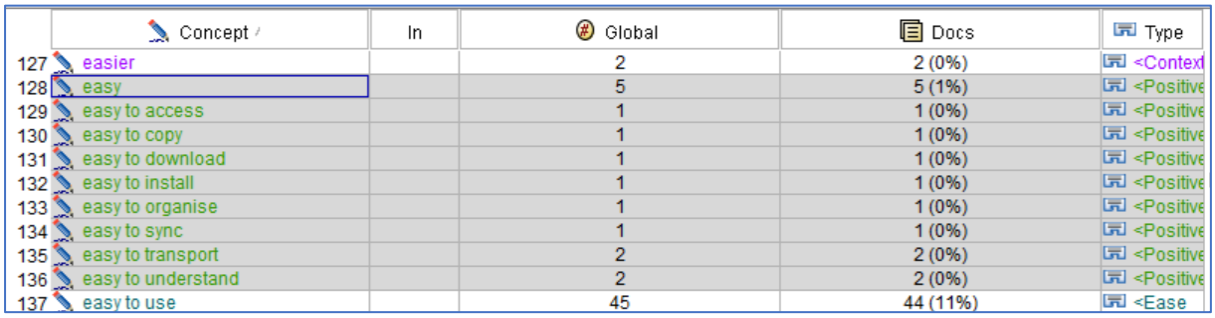
This is because we used the **Entire and Any** match method when we created the type group. This allows the term **portable** to be extracted on its own, or as part of a compound term. If we sort the concepts alphabetically by clicking on the column header, we can see where the missing occurrences have gone (Figure 6.29).

	Concept	In	Global	Docs	Type
301	pocket		2	2 (0%)	<Unknov
302	portable		36	35 (9%)	<Portabl
303	portable cassette player		1	1 (0%)	<Portabl
304	portable device		1	1 (0%)	<Portabl
305	portable keyboard		1	1 (0%)	<Portabl
306	portable music device		1	1 (0%)	<Portabl
307	portable player		1	1 (0%)	<Portabl
308	portable speakers		1	1 (0%)	<Portabl
309	powerful		1	1 (0%)	<Positive

Figure 6.29 Concepts containing the term 'portable'

If we now changed the match method in the type properties for this type group to **Entire (No Compounds)**, we would see only one instance of **portable** occurring 43 times in 42 records. Whichever match method you choose, is to a certain degree, a matter of choice.

Sorting the data in order of the concepts is another useful technique for identifying separate concepts that should belong to the same type group. As Figure 6.30 shows, if we scroll back up to the concept **easy to use**, we can see that there are many concepts that also start with the word **easy**. Maybe we should assign these concepts to the same type group as well? Or maybe we should just alter the type group so that it captures every instance of the word **easy**? Again, this depends on whether or not the concept is sufficiently meaningful, and how we intend to use it, or variations of it, when creating categories later on.



	Concept	In	Global	Docs	Type
127	easier		2	2 (0%)	<Context
128	easy		5	5 (1%)	<Positive
129	easy to access		1	1 (0%)	<Positive
130	easy to copy		1	1 (0%)	<Positive
131	easy to download		1	1 (0%)	<Positive
132	easy to install		1	1 (0%)	<Positive
133	easy to organise		1	1 (0%)	<Positive
134	easy to sync		1	1 (0%)	<Positive
135	easy to transport		2	2 (0%)	<Positive
136	easy to understand		2	2 (0%)	<Positive
137	easy to use		45	44 (11%)	<Ease

Figure 6.30 Concepts beginning with the word 'easy'

Returning to the Filter dialog, we can look at concepts in the **unknown** type group that might be added to our two local types. Once again, click:

Within the dialog, click the radio button:

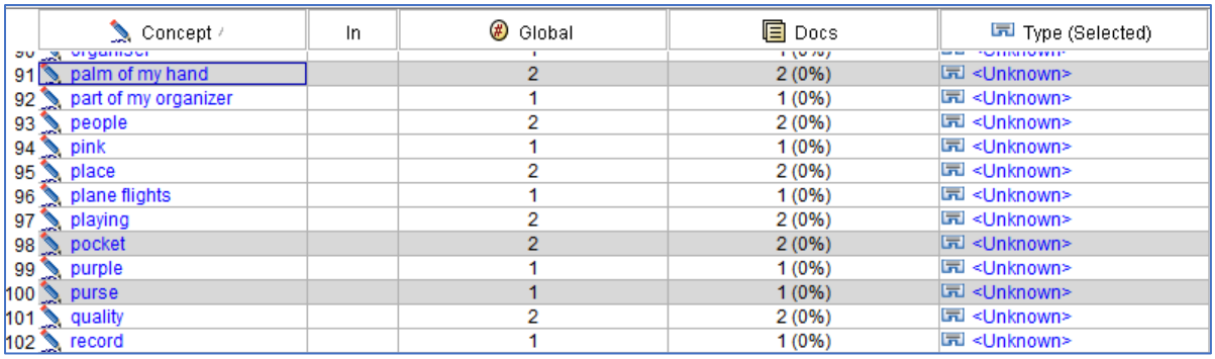
Selected Types

From the list of types, choose:

<Unknown>

If we again sort the concepts in alphabetical order, and scroll downwards until we able to select the following terms:

***Ctrl-Click to select the concepts* palm of my hand, pocket *and* purse**



	Concept	In	Global	Docs	Type (Selected)
91	palm of my hand		2	2 (0%)	<Unknown>
92	part of my organizer		1	1 (0%)	<Unknown>
93	people		2	2 (0%)	<Unknown>
94	pink		1	1 (0%)	<Unknown>
95	place		2	2 (0%)	<Unknown>
96	plane flights		1	1 (0%)	<Unknown>
97	playing		2	2 (0%)	<Unknown>
98	pocket		2	2 (0%)	<Unknown>
99	purple		1	1 (0%)	<Unknown>
100	purse		1	1 (0%)	<Unknown>
101	quality		2	2 (0%)	<Unknown>
102	record		1	1 (0%)	<Unknown>

Figure 6.31 Selecting multiple unknown concepts in the extraction pane

We can simultaneously add them to the type group **portable**. By right-clicking and selecting from the pop-up menu:

Add to type

Portable

Figure 6.32 shows this process in action.

As soon as this is done, we will need to re-extract the data. Having done so and having returned to the unfiltered view of the extraction pane, the re-extracted concepts are shown in Figure 6.33 with the viewing mode changed so that the type groups are displayed. We can now see that the **portable** type accounts for 48 concepts in 46 records.

Of course, users may prefer to only re-extract the concepts after they have done this for a number of unassigned concepts, as extraction takes time to complete.

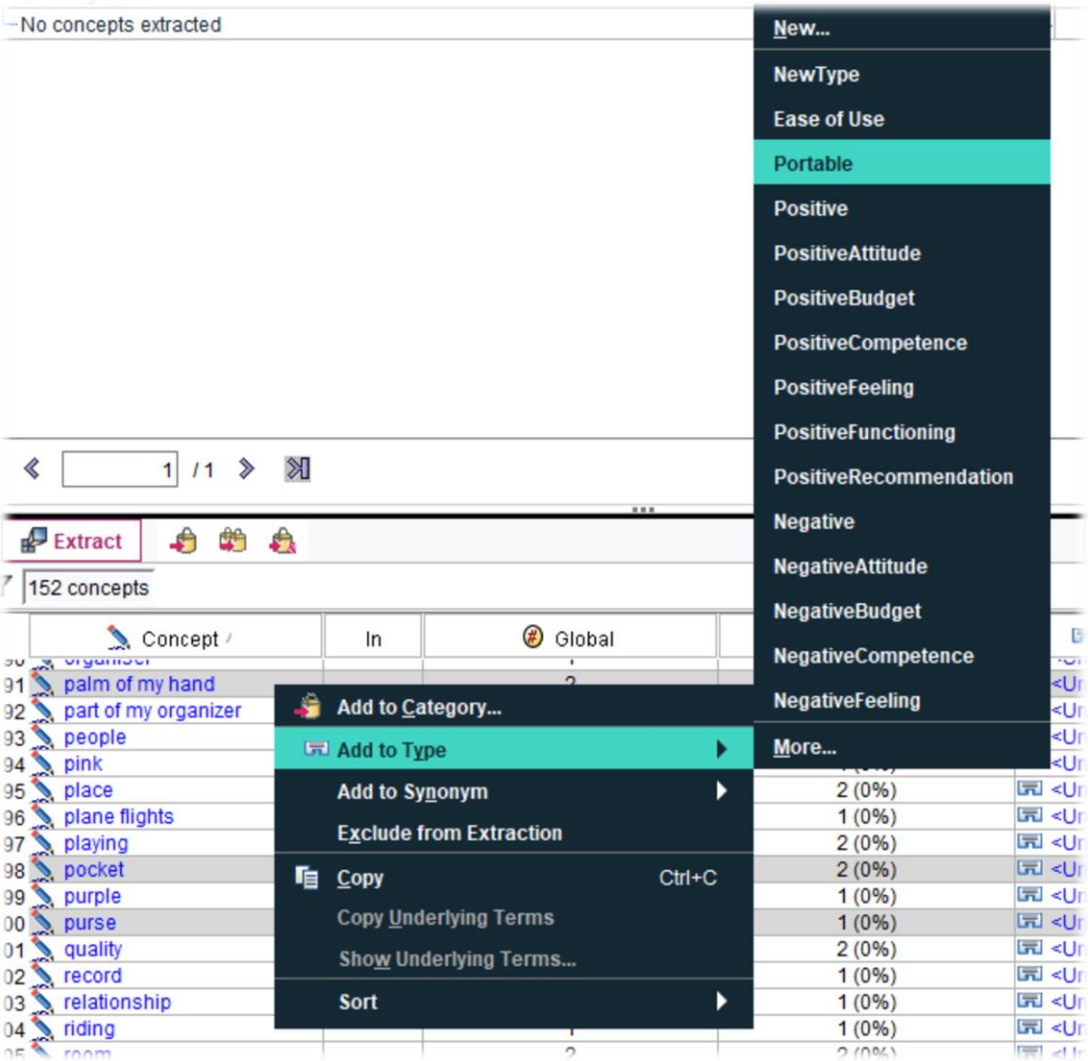


Figure 6.32 Adding multiple unknown concepts to the type group 'portable'

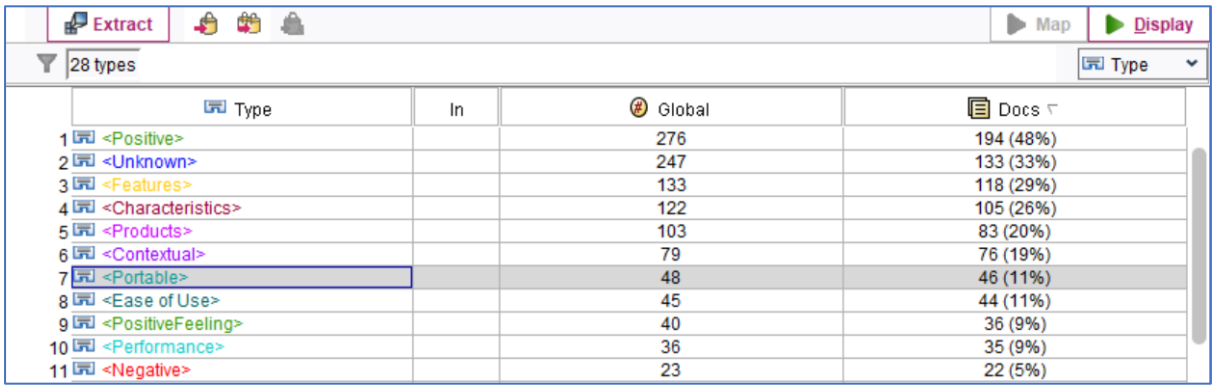


Figure 6.33 The updated 'Portable' type group in the extraction pane

With regard to managing and sharing resources, you should know that the Resources menu in the resource editor allows user to export entire Resource templates as individual files (with the extension **.lrt**) so that they may be shared with others. The same can be done with individual library files (with the extension .**lib**). To show how we can use these files, we can import a previously created and more complete version of the local library we have been creating in this chapter. From the main menu click:

Resources

Manage Libraries

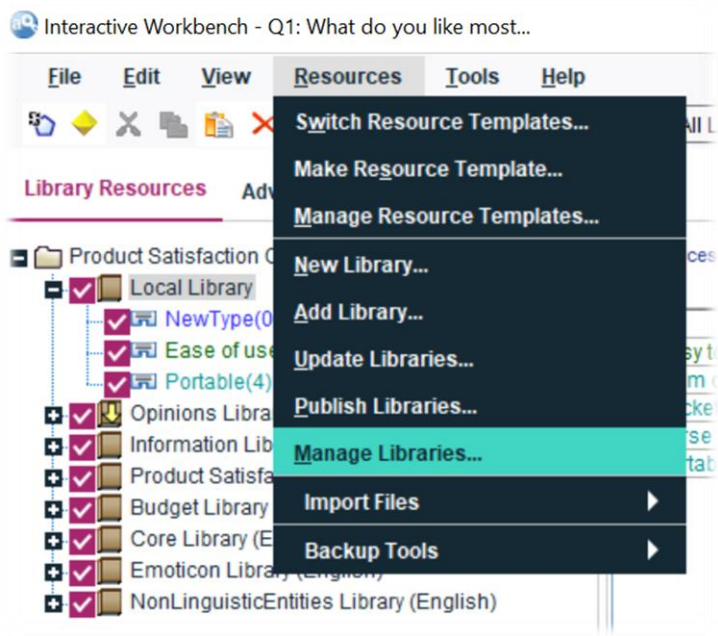


Figure 6.34 Location of the Manage Libraries dialog in the resource editor

The Manage Libraries dialog appears as shown in Figure 6.35

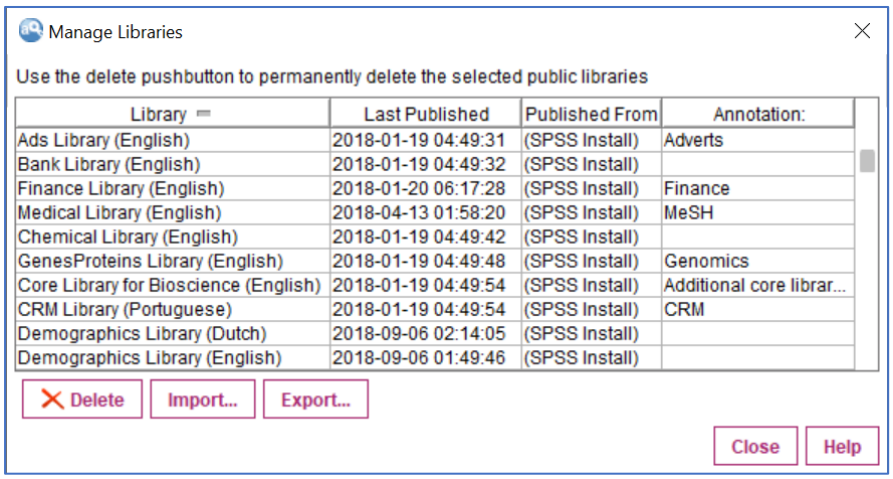


Figure 6.35 The Manage Libraries dialog

Within the dialog, click:

Import

Browse to the data folder (in this example it can be found in **C:\SV Training\Text Mining\Data**) and select the file:

MP3.lib

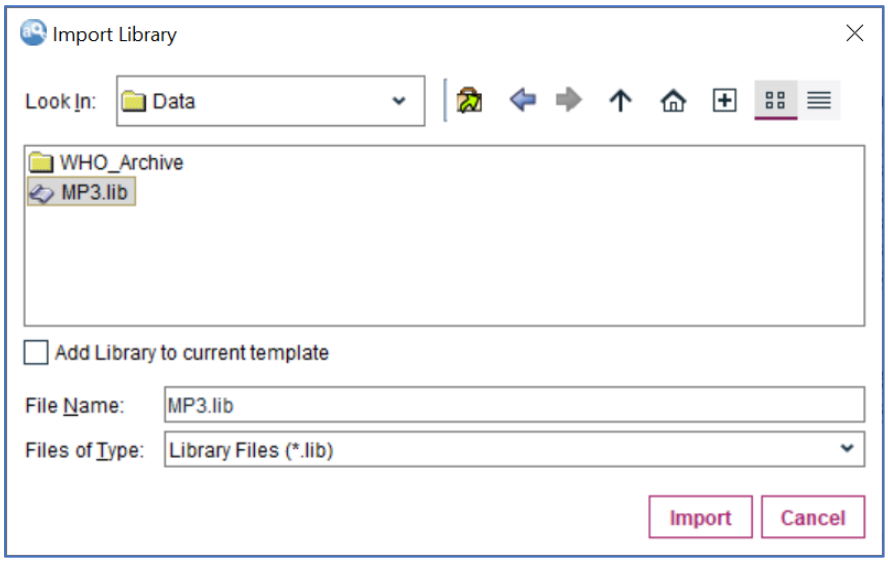


Figure 6.35 The Import Library dialog

Figure 6.36 shows the import process.

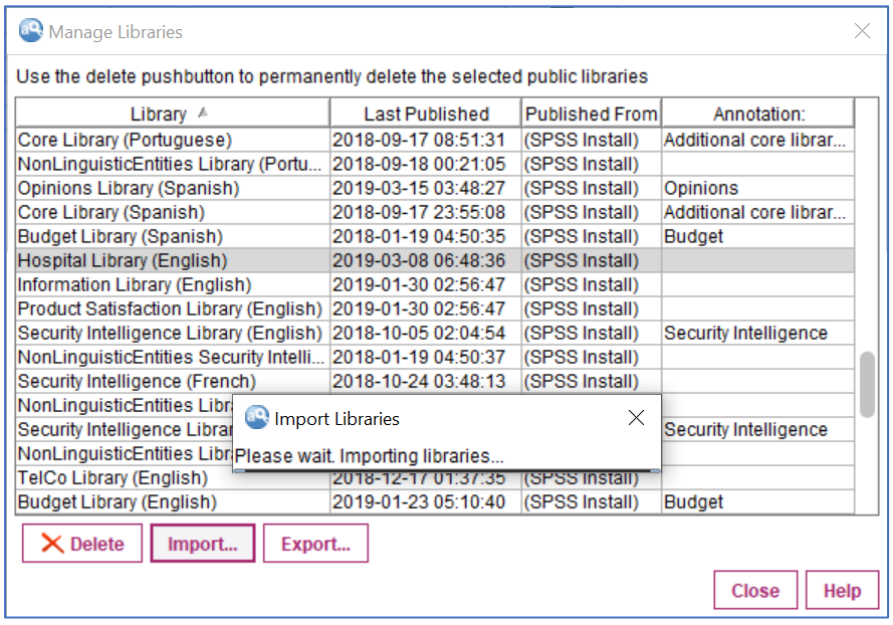


Figure 6.36 Importing the MP3.lib library file

When the import is complete, click:

Close

The library has now been imported to the system's internal repository of libraries files, so we can add it to the existing resource template. To do so, click:

Resources

Add Library

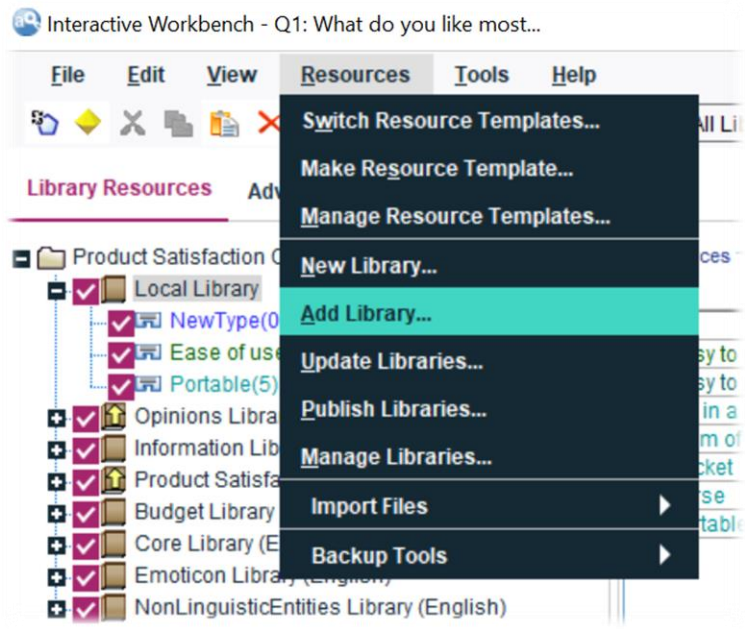


Figure 6.37 Adding a library to the existing resource template

Find the imported **MP3** library and click:

Add

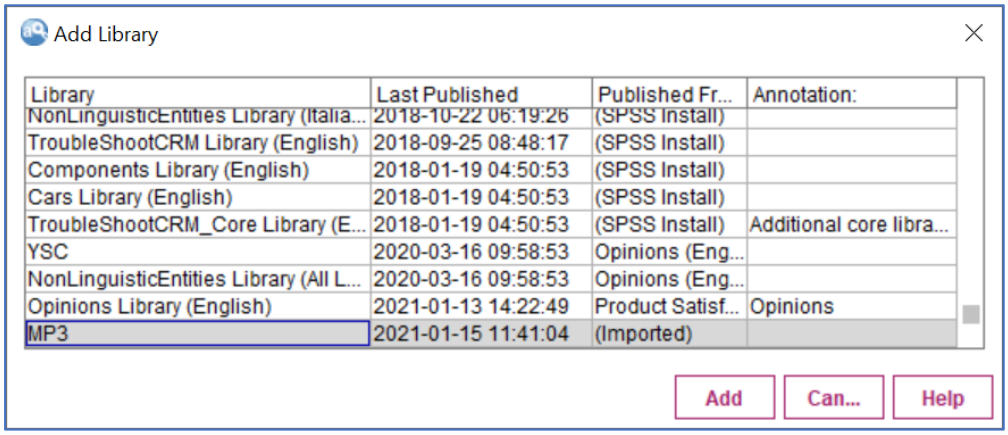


Figure 6.37 Adding the MP3 library to the existing resource template

As Figure 6.38 shows, once again we encounter an **Edit Forced Terms** dialog. This is simply because the MP3 library contains terms that already exist in both our local library and the Opinions library.

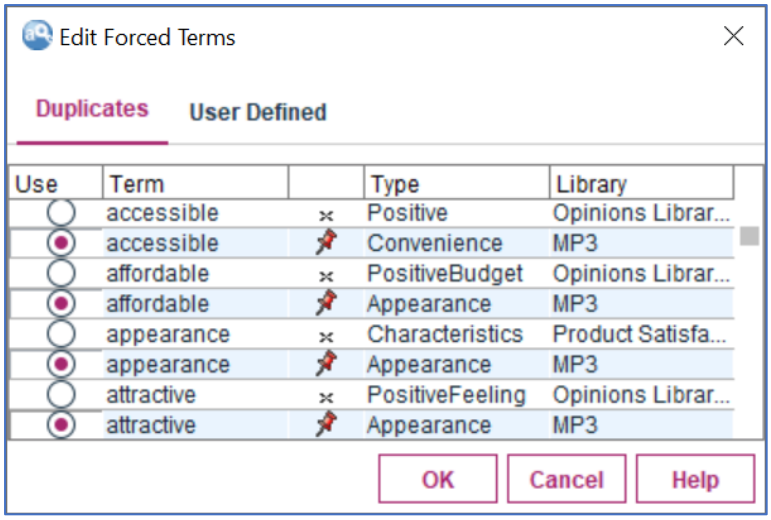


Figure 6.38 Edit Forced Terms dialog: This appears as the MP3 library contains terms that are already present in the local and Opinions libraries

As we wish to reassign these terms to new type groups in our imported MP3 library, we can continue by clicking:

OK

Figure 6.39 Library folders in the resource editor after the MP3 library has been imported

Within the local library:

Select the two type groups* Ease of use *and* Portable *and delete them

We can now return to the categories and concepts window and re-extract the concepts taking account of the updated resource files. Figure 6.40 shows the extraction pane using the type view. Note the presence of many new type groups such as **Capacity**, **Appearance**, **Design** and **Battery**.

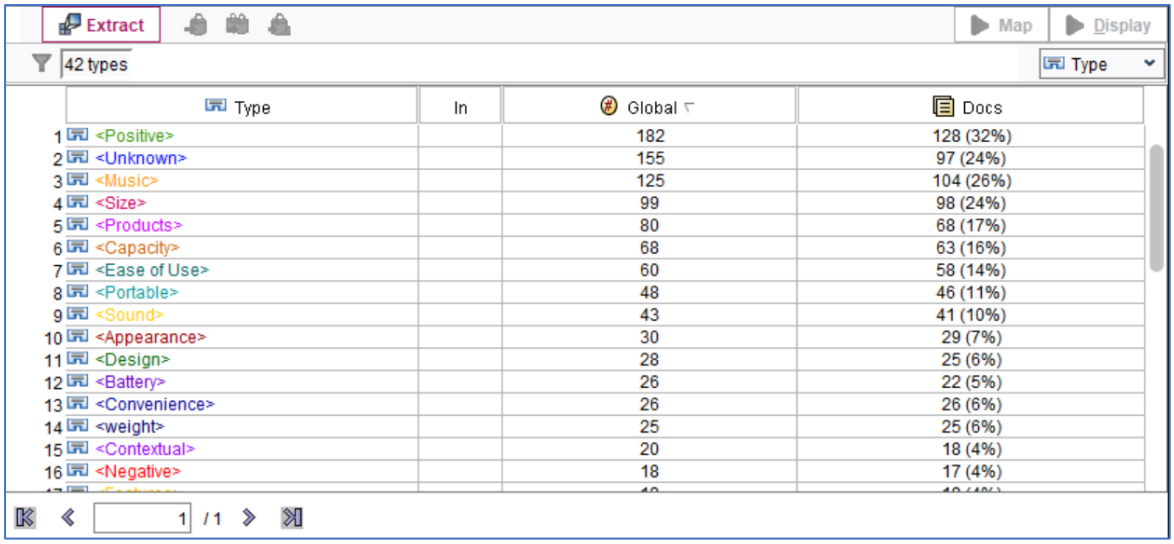


Figure 6.40 New type groups in the extraction pane as a result of importing the MP3 library

To finish working on this instance of the interactive workbench click:

File

Close

From the **Close Interactive Session** dialog choose:

Update

From the **Save and Exit** dialog, choose the second option:

Keep session work and cache text data with extraction results for reuse

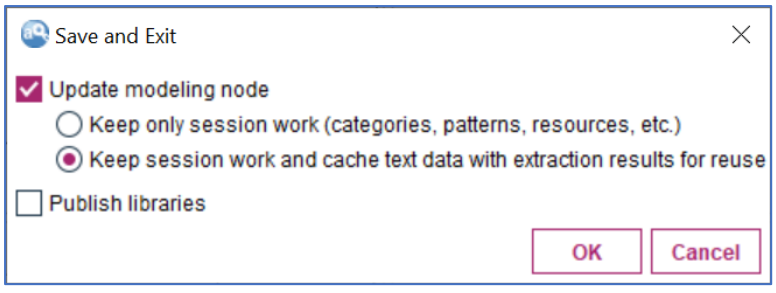


Figure 6.41 Updating the modelling node with session work and cached text data

When the interactive workbench closes, save the stream with a new name. From the main menu click:

File

Save Stream as…

06_Working_With_Resources_Complete.str

Save

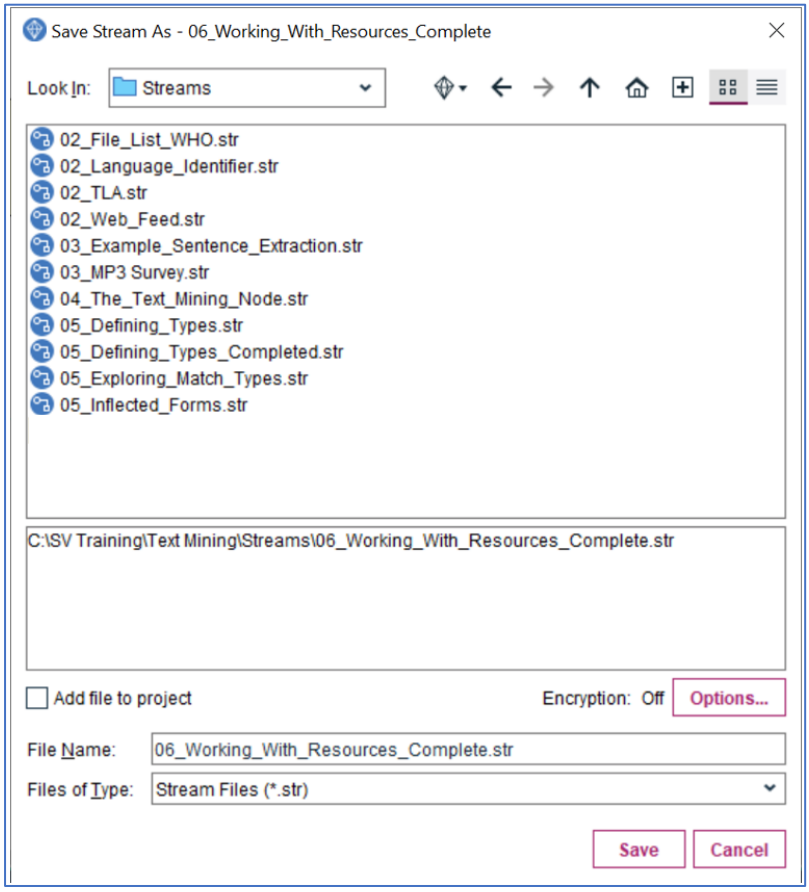


Figure 6.42 Saving the stream with a new name

Having saved the stream with the session work and the extracted data cached in the text mining modelling node, we will be able to continue where we left off when we re-run the stream later.

Practice Exercise – Chapter 6

Within the folder **Student Exercises** open the following stream:

Chapter_06_Practice.str

1. Right-click on the text mining node and change the loaded resource template from **Basic Resources (English)** to **Opinions (English)**. Now, run the text mining node.

2. Once the extraction has completed, notice the number of concepts that are still typed as **unknown**. To improve the extraction results and to illustrate how we can change the template after the interactive session has started, switch to the **Resource Editor** view, and from the **Resources** menu, switch the resource template to **Customer Satisfaction Opinions (English)**. Return to the **Categories and Concepts** window and after again extracting the concepts, notice that concepts such as **Customer Service** have now been typed as part of the **Customer Support** type group.

3. Let's further improve the results by adding a library of terms specifically created for the car rental sector. Return to the **Resource Editor** view and within the **Resources** menu, click **Manage Libraries**. Use the resultant dialog to navigate to the **data** folder in **C:\SV Training\Text Mining\Data** and import the library file **Car Rental.lib**. After importing it, close the dialog, return to the **Resources** menu and click **Add Library**. In the resultant dialog, find and add the **Car Rental** library to the current resource template. Click **OK** at the **Edit Forced Terms** dialog and return to the **Categories and Concepts** window before once again re-extracting the concepts.

4. The extraction results show that many key concepts have been typed under generic type groups such as **Positive** or **Positive Attitude** or **Negative**. Let's assume we wish to further enhance our **Car Rental** library using some of these already typed concepts. Return to the **Resource Editor** view, and right-click on the **Car Rental** library to add a type group. Create two new type groups as follows. In each case, the terms are already associated with an existing type. Make sure you force them so that they are associated with their *new* type group.

Type Name	Match	Inflected	Colour	Terms
fast	Entire and Any	Yes	green	fast
slow	Entire and Any	Yes	red	slow, too long

Following re-extraction, the concepts added to the new type groups match with about 22 cases. This is partly because they are also the target terms for several synonyms as defined in the substitution dictionary. This approach, whereby we reassign concepts to new type groups, may help us to create templates that contain type groups that refer to more specific aspects of the text data: in this case speed of service.

If you have time, feel free to reassign a few more concepts to new type groups in the **`Car Rental`** library.

5. Finally, we can use this opportunity to create a new template. To do so, return to the **`Resource Editor`** and from the **`Resources`** menu choose **`Make Resource Template`**. Call the new resource template **`Car Rental`**.

6. Exit the interactive workbench without updating the node.

Chapter 7 Creating Categories

With most text mining applications, the ultimate goal is to classify the contents of the text data in some fashion. In Modeler Text Analytics, this most commonly takes the form of creating categories. There are several methods that users can employ to create text categories.

1. **Concepts** – Right-clicking on extracted concepts within the extraction pane allows users to create a new category or add a concept to an existing category. Concepts may also be dragged and dropped into existing categories.
2. **Types** – Using the type view within the extraction pane, users can also right-click on a type group and choose to create new categories or to add the type to an existing category. Types groups may also be dragged and dropped into existing categories.
3. **Category Rules** – Category rules are syntax-based statements that assign documents or records to categories using a logical expression. Each rule is a descriptor of a single category. An example of a rule might be one which assigns all mentions of the word **store** to the **capacity** category except when type group **purchase** occurs in the same record.
4. **Automatic Categorisation** – Modeler Text Analytics also contains methods for automatically generating categories. Using linguistic and frequency algorithms, this approach can produce categories from extracted concepts or concept patterns.
5. **Text Link Analysis** – Text Link Analysis creates patterns of co-occurrence between concepts and/or type groups. For example, TLA might generate a descriptor pattern that identified instances of phrases containing concepts from the type group **sound** that co-occurred with concepts in the **positive** type such as ***great sound quality***.
6. **Import Predefined Categories –** Lastly, it is possible to import files in Excel format with predefined categories. The software supports files with different arrangements of categories, whether flat or hierarchical. It also includes the ability to autodetect the file structure.

7.1 Creating categories from concepts

Running the stream contents in **07_Creating_Categories.str** allows us to once again extract concepts from the previous chapter's data file and utilise the MP3 dictionary of type groups. When the extraction process is complete, we can return to the concepts in the extraction pane. This time, we will look at the process of creating categories. There may be occasions when a concept refers to something quite specific within its type group. For example, the type group **size** contains the concepts

small and **size**. When asked to indicate what they like most about the mp3 player, many respondents have mentioned that that they like the product's size or that specifically like its *small* size. If we wish to create a category that just contains the concept **small**, we can do so simply by right-clicking on the concept and selecting the appropriate procedure from the pop-up menu. But first, let's remind ourselves that this concept also refers to a number of specific underlying terms. Within the extraction pane select the concept:

small

Right-click on the concept and from the pop-up menu select:

Show Underlying Terms

Figure 7.1 shows the resultant window of underlying terms.

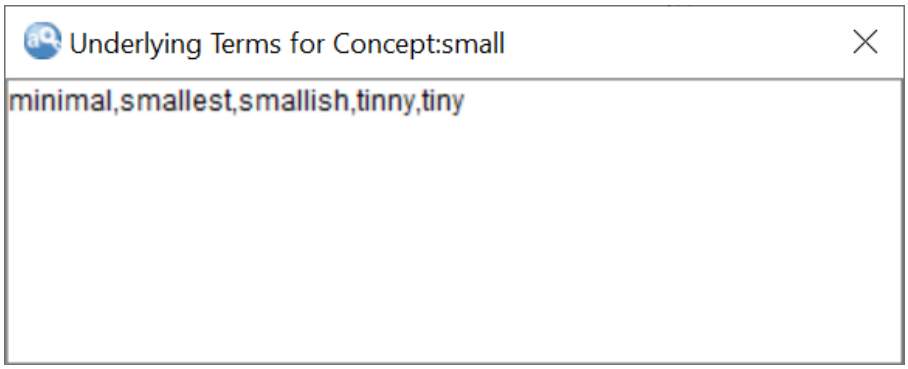


Figure 7.1 Underlying terms for the concept 'small'

Having reminded ourselves that this concept also refers to words other than the single word **small**, we can dismiss the window and again right-click on the concept:

small

From the pop-up menu, shown in Figure 7.2 select:

Add to Category

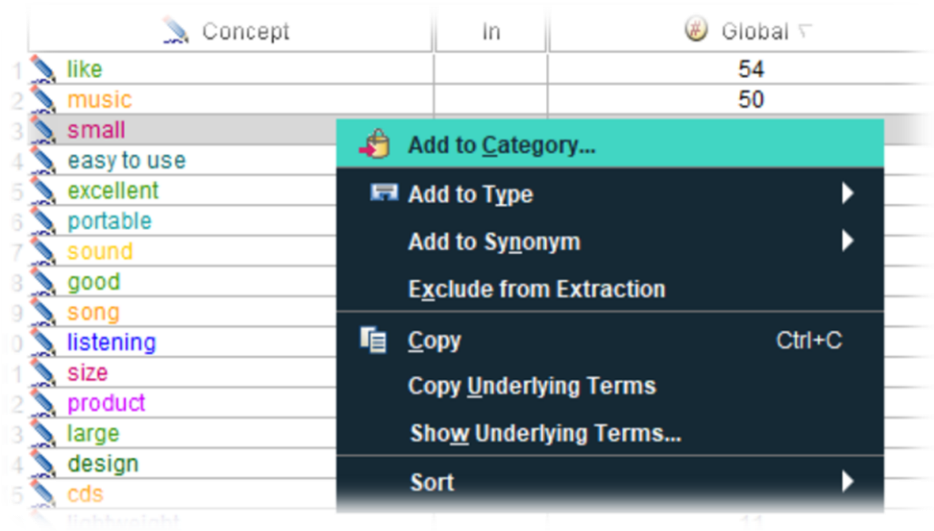


Figure 7.2 Adding a concept to a category

As Figure 7.3 illustrates, no categories have yet been defined, so the only option offered is to select:

Create New Category

And click:

OK

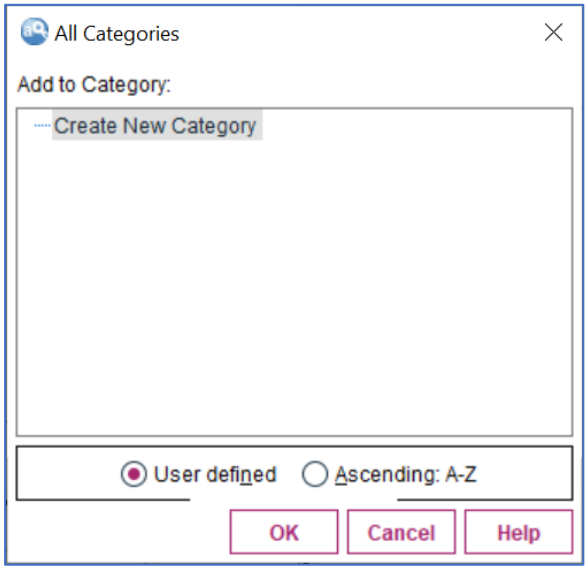


Figure 7.3 Creating a new category

We can now see that a new object appears in the category pane of the interactive workbench. The new category is represented by an icon in the shape of a bucket. The adjacent value in the column marked **Descriptors** indicates that this category contains only one descriptor: the concept **small**.

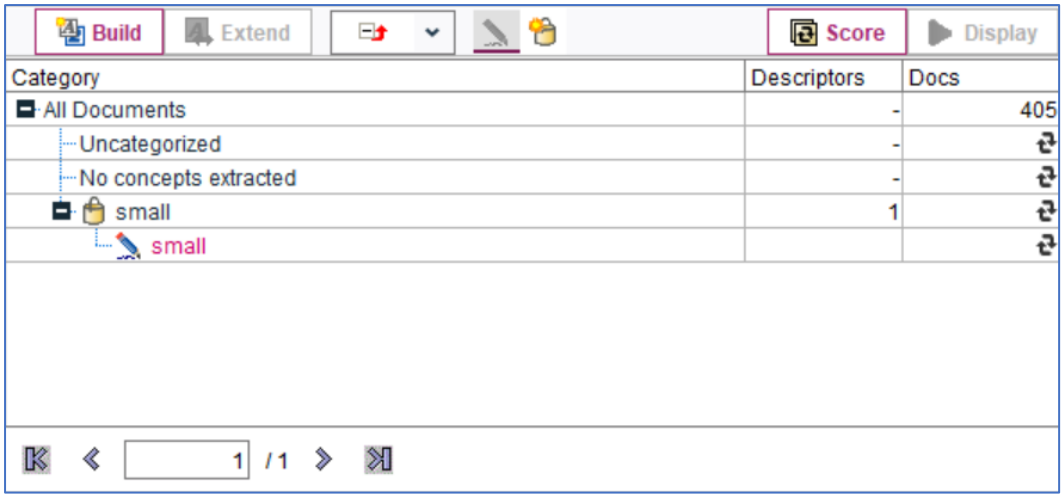


Figure 7.4 The category pane showing the newly created category 'small'

If we want to see how many records a given category contains, we need to *score* the data. To do so click the button marked:

Score

The scoring process now begins whereby each record is evaluated to see if they match the descriptors contained in any existing categories.

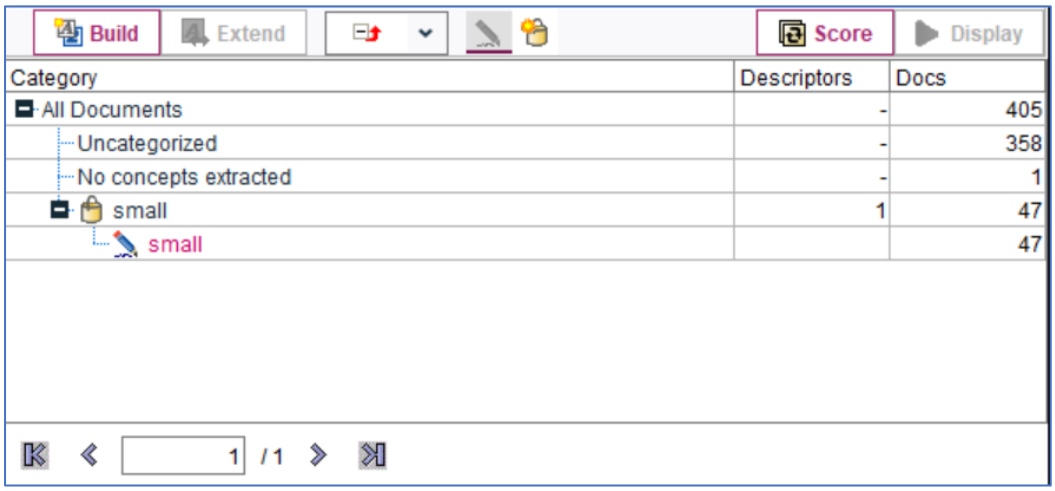


Figure 7.5 The category pane showing scored results

The category pane now shows that there are 405 docs (or records) available for analysis, as indicated by the row header **All Documents**. It also shows that there are 358 records remaining that are yet to be categorised, as indicated by the row header **Uncategorized**. It further indicates that there is one record where the parts of speech pattern rules resulted in **No concepts extracted**, before finally showing that the category **small** contains 47 records, all of which are associated with the single descriptor **small**. We can view any of these groups of records by selecting the

appropriate cell and clicking the **Display** button. To show the 47 records in our single category, click the category row:

small

Now click the button marked:

Display

Figure 7.6 shows the data pane displaying the 47 records with highlighted terms.

		Q1: What do you like most about this portable music pl	Categories
1	89.0	...Small but lots of space (60 GB). Video is a bit of a toy but cool....	small
2	265.0	...It's small...	small
3	97.0	...It's very small....	small
4	185.0	...The size. It can take a beating and any jarring won't make the music skip at all. Also, it is small...	small
5	105.0	...Small and easy to use...	small
6	342.0	...It's small so no one notices that I'm listening to it at work....	small
7	196.0	...Small...	small
8	348.0	...small and neat, easy to take around with me...	small
9	96.0	...its tiny and can hold lots of songs and photos...	small
10	83.0	...It's small & fits in your pocket....	small
11	263.0	...It's nice and small and fits into my handbag....	small
12	183.0	...I can listen anywhere and I can plug into my speaker system. It is small and i can play music...	small

Figure 7.6 The data pane displaying records belonging to the category 'small'

Meanwhile, the extraction pane shows a category bucket icon has appeared next to the concept **small** indicating that is has been used as a category descriptor.

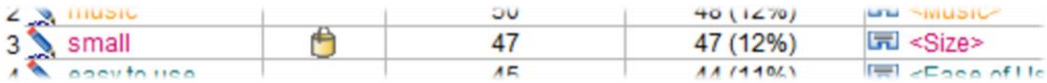


Figure 7.7 The extraction pane showing that the concept 'small' has been used as a category descriptor

An alternative approach would be to create a hierarchical categorisation schema that would encapsulate the category **small** as a subcategory of **size**. To show this, right click on the concept:

size

from the pop-up menu, once again choose:

Add to Category

Create New Category

The category **size** is now added to the category pane (see Figure 7.8).

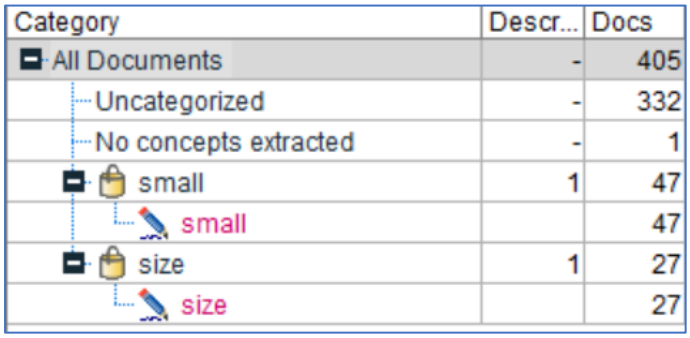


Figure 7.8 The category 'size' added to the category pane

We can now move the category **small** so it becomes a subcategory of **size**. To do so, right click on the category:

small

From the pop-up menu, choose:

Move to Category

Now the **Move to Category** window opens (see Figure 7.9). From the category tree choose:

size

Click:

OK

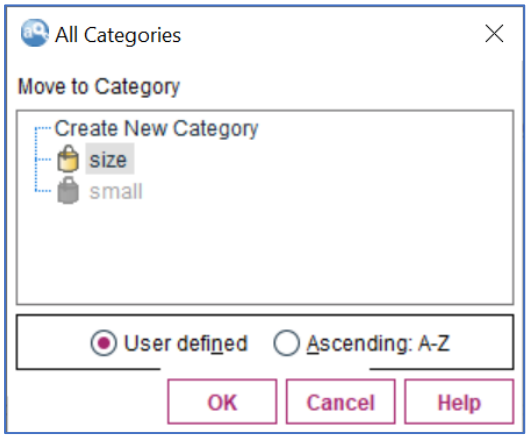


Figure 7.9 Move to Category window

As Figure 7.10 shows, the **size** category now captures 73 records as it also contains the subcategory **small**. You should note that we can also drag and drop categories in the category pane to create hierarchies manually.

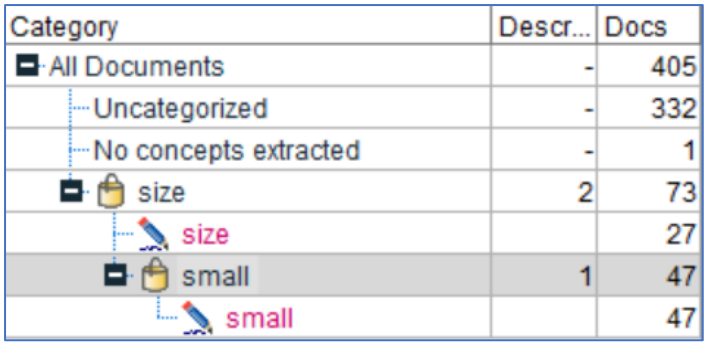


Figure 7.10 Hierarchical categories in the category pane

If we later decide that having hierarchical categories is unnecessary, we can *flatten* them within the category pane. To illustrate, right-click on the category

size

From the pop-up menu choose:

Flatten Categories

As Figure 7.11 shows, the **size** category now consists of two concepts with no subcategories.

Figure 7.11 Flattened category structure

7.2 Creating categories from types

Often it more efficient to simply create categories from types rather than concepts. Indeed, one of the main reasons we go to so much trouble to create appropriate type groups in the first place, is because they are so useful for categorisation purposes.

To illustrate, if we sort the concepts in alphabetical order by clicking on the column header marked **Concept**, and then scroll down to the word **easy**, we can see that there are several compound terms beginning with this word, all of which belong to the type group **Ease of Use** (see Figure 7.12).

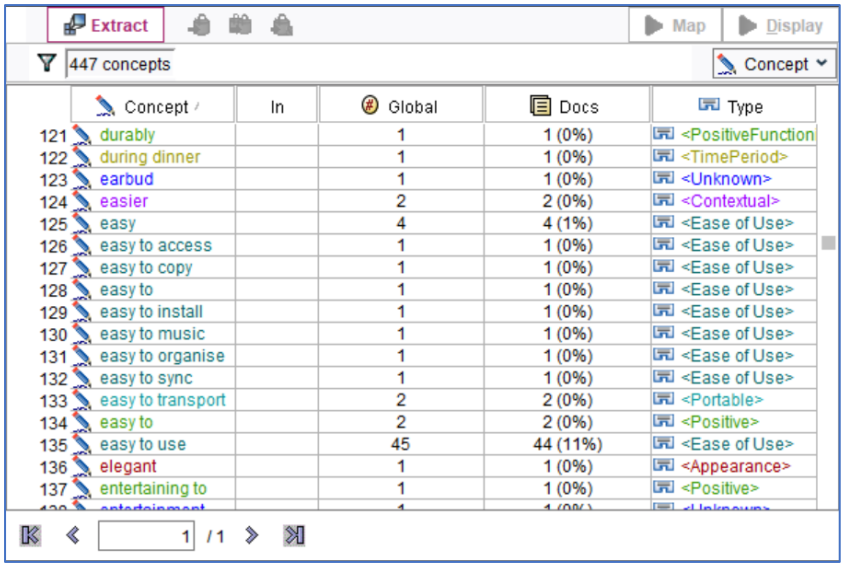


Figure 7.12 The extraction pane sorted in alphabetical order showing concepts beginning with the term 'easy'

If we switch the extraction pane view to **Type**, we can simply right-click on the type group **Ease of Use** and create a new category based on it or add it to an existing category. Figure 7.13 illustrates the process of creating a new category from this type group.

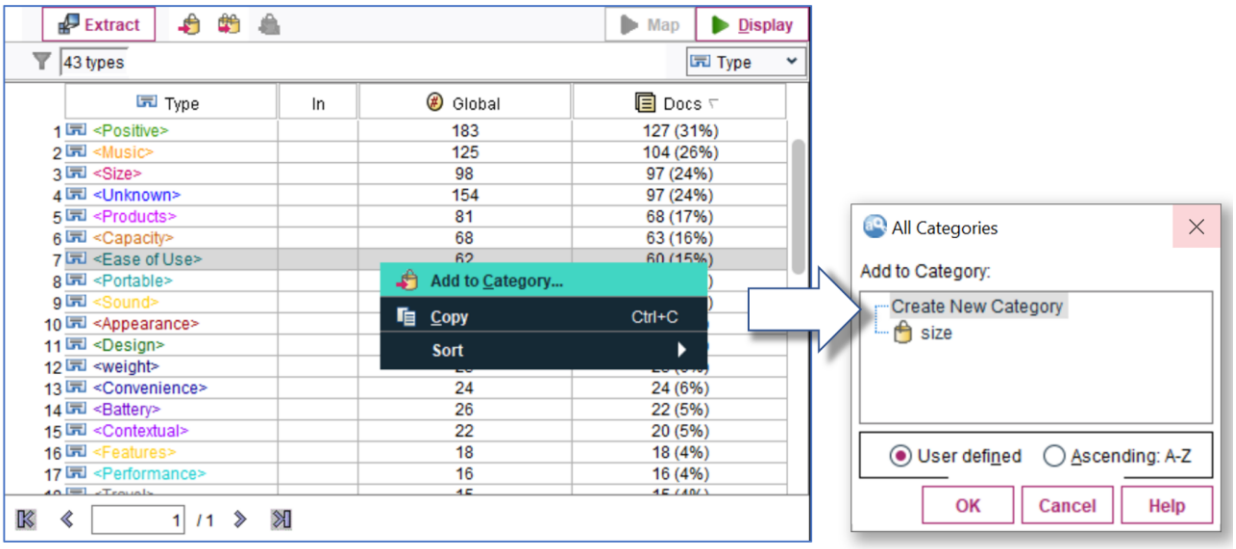


Figure 7.13 Creating a new category from the type group 'Ease of use'

You may notice the use of **<>** symbols that encapsulate both the category name and the descriptor. This indicates that a type group has been used rather than a concept to create the category.

Category	Descriptors	Docs
All Documents	-	405
Uncategorized	-	317
No concepts extracted	-	1
size	2	73
size		27
small		47
<Ease of Use>	1	19
<Ease of Use>		19

Figure 7.14 Creating a category based on the type 'Ease of use'

If we continue this process, we can add new categories based on the following type groups.

Appearance (including the type group **Colour**)

Battery

Capacity

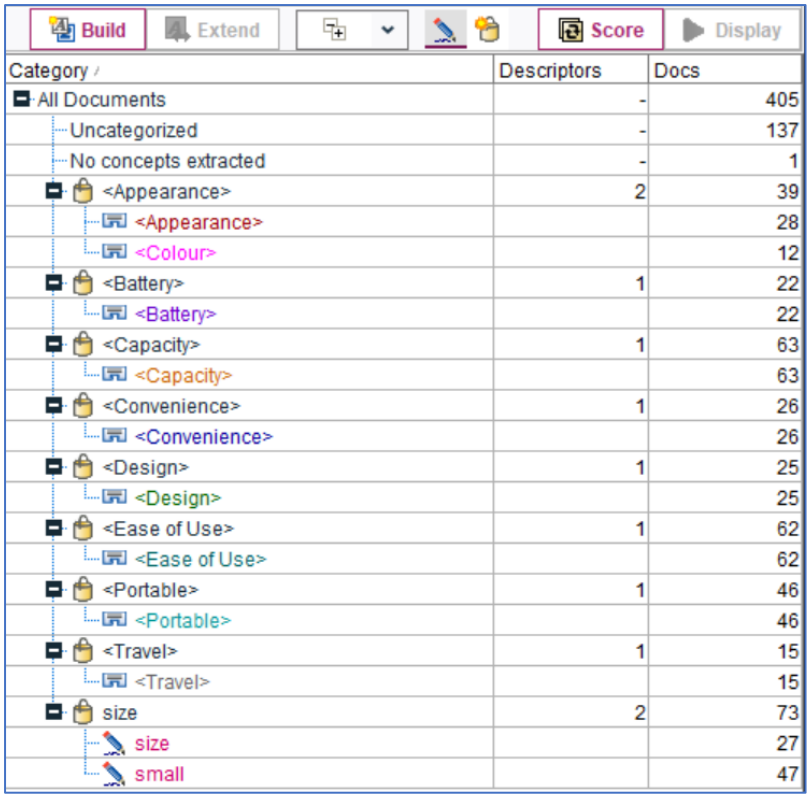
Convenience

Design

Portable

Travel

Figure 7.15 shows these additional category groups added to the category panel.



Category	Descriptors	Docs
All Documents	-	405
Uncategorized	-	137
No concepts extracted	-	1
<Appearance>	2	39
<Appearance>		28
<Colour>		12
<Battery>	1	22
<Battery>		22
<Capacity>	1	63
<Capacity>		63
<Convenience>	1	26
<Convenience>		26
<Design>	1	25
<Design>		25
<Ease of Use>	1	62
<Ease of Use>		62
<Portable>	1	46
<Portable>		46
<Travel>	1	15
<Travel>		15
size	2	73
size		27
small		47

Figure 7.15 Additional categories created from types

Figure 7.16 shows that you can choose to display the descriptors within the categories, or just the categories themselves, with the expansion control buttons at the top of the category panel.

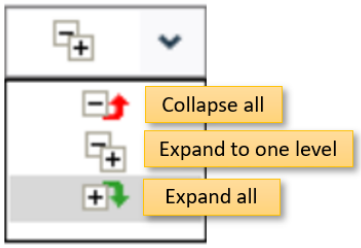


Figure 7.16 Expansion control buttons in the category pane

7.3 Creating categories with rules

Category rules allow users to define relationships between types, concepts and patterns that may co-occur within a document/record. Category rules can be generated automatically or created manually within Modeler Text Analytics. To illustrate, we will create an empty category within the category pane. Right-click at the top the category pane and from the drop-down menu choose:

Create Empty Category

Within the category properties dialog give it the name:

Online

Figure 7.17 illustrates this process.

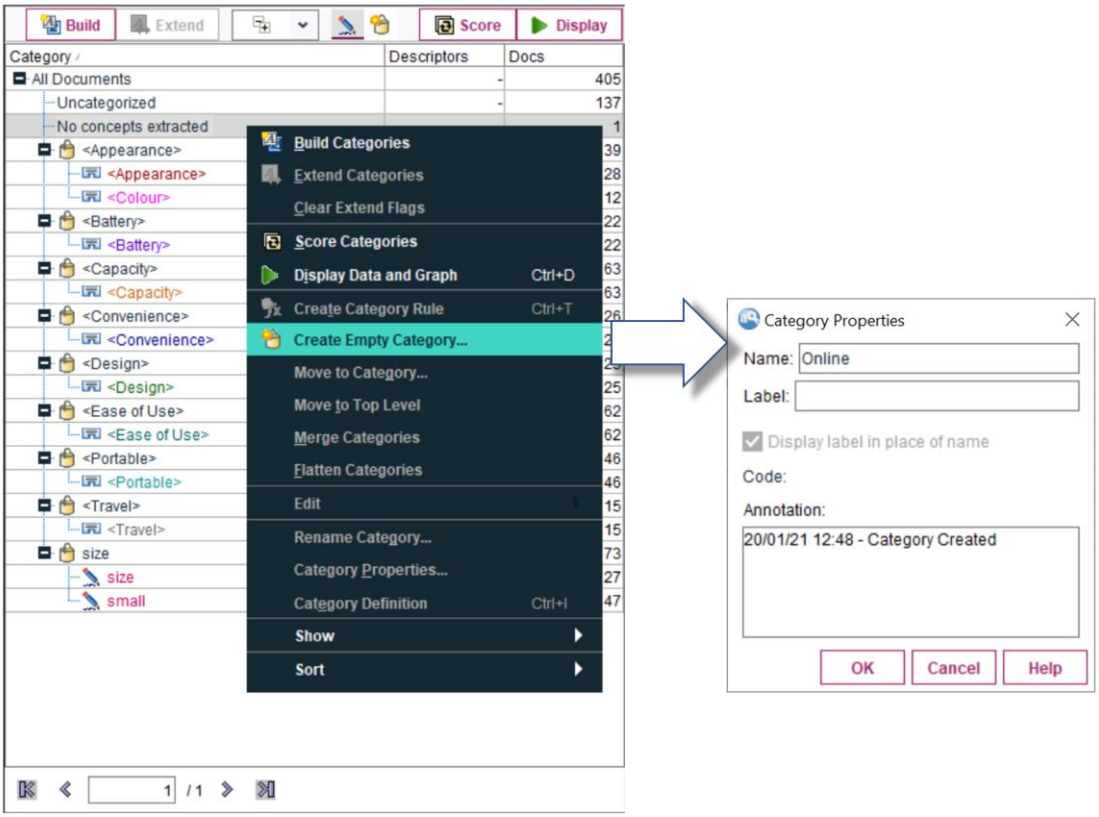


Figure 7.17 Creating an empty category and naming it 'Online'

Now, right-click on the new category, and from the drop-down menu choose:

Create Category Rule

The system now generates the category rule editor as shown in Figure 7.18.

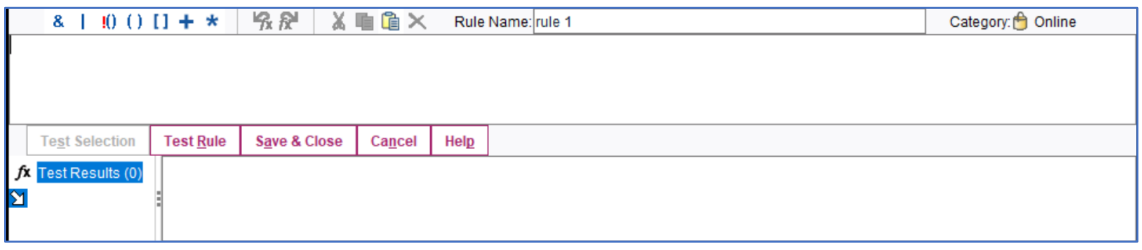


Figure 7.18 The category rule editor

To build up category rules, users may drag and drop actual concepts, types and patterns directly into the rule editor. When doing so, each of these items appears with its own associated icon:

 Extracted Concept

 Extracted Type

 Extracted TLA Pattern

As with any rule editor of this type, there are a number of operators that are used to form the basic syntax of a rule. Figure 7.19 lists these along with a short description.

Operator	Description
&	This is the "*and*" boolean. For example, **item1 & item2** defines categories that contain both items such as: **complaint & service**
|	The "*or*" boolean acts as an inclusive operator, meaning that records are counted if any or all of the elements are found. For example, **item1 | item2** defines categories that contain either item: **rental** or **hire**
! ( )	The "not" boolean is a negation operator that helps defines categories in terms of what they *don't* match with. For example, **!(item1)** means the category should not contain the item. This is useful when working with ambiguous terms. For example, if the term **bank** is only supposed to refer to financial concepts, we could employ the condition **!(river)** to avoid any records that refer to waterways.
*	This is a wildcard symbol which is useful when working with compound terms. For example, ***item1** means the category with match with any terms containing item1 at the end of a word. So ***phone** will match with **iphone** whereas * **phone** (with a space between * and **phone**) will match with **mobile phone**, **cellular phone** and **home phone**.
( )	This is an expression delimiter. It means that any expression within the parentheses is evaluated first.
+	This operator is a pattern connector. It's used to specify the order of items within Text Link Analysis (TLA) pattern. It needs to be used in conjunction with square brackets.
[ ]	This is the pattern delimiter that is used when matching data based on an extracted TLA pattern. Any content within the brackets is expected to be a TLA pattern, so it doesn't work with concepts or types based on simple co-occurrences within the record.

Figure 7.19 Syntax operators for category rules

It's generally advisable to drag and drop concepts or types directly into the editor, but in this example, we can simply type an expression that uses the **or** boolean to match with terms related to online topics. Within the editor type:

internet | web | online

Figure 7.20 shows this in the editor.

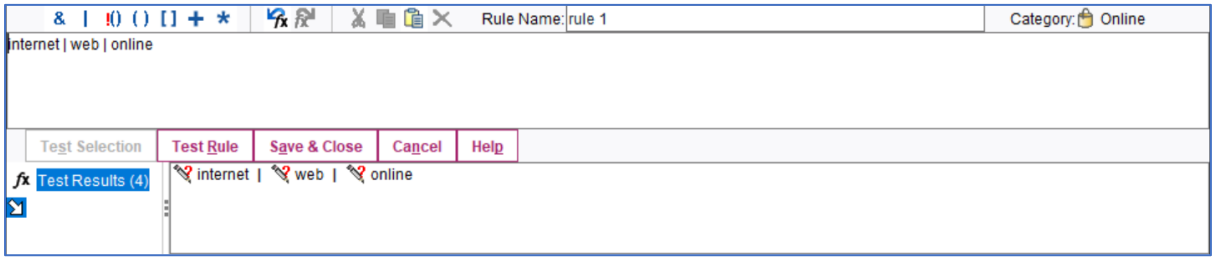


Figure 7.20 Category rule editor with expression using the 'or' Boolean

We can test how many records the rule will match with by clicking the button marked:

Test Rule

Looking at the results in the data pane, the rule matched with four records. However, none of these records contain the term **online** (see Figure 7.21). That is because the word **online** appears as part of a compound term.

		Q1: What do you like most about this portable music pl	Categories
1	26.0	...its great. i can share music with my friends and download tons of tunes off the internet....	Online
2	323.0	... and thinner and holds like 15,000 songs. For me though, having access to tunes on the web...	Online
3	151.0	..., and I can run other programs, take notes, as well as listen to music. If I wanted to, I could surf the Web...	<Capacity> Online
4	272.0	...Creating custom playlists, listening to comedy clips downloaded from the Web...	Online

Figure 7.21 Records matching with the currently defined category rule

Let's edit the category rule so that we can include a wildcard (*) to see if we can capture mentions of the term **online**. Edit the rule so that it now looks like this:

internet | web | online *

If we re-run the **Test Rule** procedure, we can now see that the data pane contains five records. The additional record includes the term **online store** (see Figure 7.22).

2	148.0	...The online store is great. Also, sound quality is excellent....	<Capacity> Online

Figure 7.22 An additional record appears in the data pane containing the compound term 'online store'

The problem here is that the concept **online store** is matching with the type group **capacity**, which is why this record is also part of the **capacity** category. This is of course, not what the type group **capacity** was meant to capture. Let's think about what options do we have to fix the problem, so that the concept **online store** no longer appears in the capacity group. Here are some potential actions we might take:

- Change the match method in the type group **capacity** from **Entire and Any** to **Entire and Start** so that the word **store** no longer matched as part of the expression **online store**. Unfortunately, this might affect other legitimate matches where the term **store** appears at the end of an expression.
- Create a type group especially for the term **online store** so that it doesn't match with capacity anymore (this would work)
- Create a rule for the category **capacity** so that it matched with all the other records in the type group **capacity** except the one containing **online store** (this will also work).

We can explore this third option by firstly saving the new category rule for online by clicking the button marked:

Save & Close

We can now return to the category pane and expand the category **capacity**. Within the category group:

Click the descriptor* category *and press the delete key

Now that the category group is empty, we can right click-on the group and select:

Create Category Rule

The first thing we need to do is to drag the type group **capacity** from the extraction pane into the rule editor (make sure the view is set to type within the extraction pane). You will see now that the expression simply contains the single type group **<Capacity>** within the editor. If we test this rule it will match with 63 records including the record with the id number **148** that contains the phrase **online store**. Now let's edit the expression so that it reads as follows:

<Capacity> & !(online store)

By adding the **not** boolean, we are requesting the rule finds records that match with the type group **capacity** but *not* any containing the concept **online store**. Figure 7.23 shows the editor results after pressing the button marked:

Test Rule

Note that the bottom left-hand corner shows the test results, indicating that the rule now matches with only 62 records as the record containing the concept **online store** no longer falls into this category.

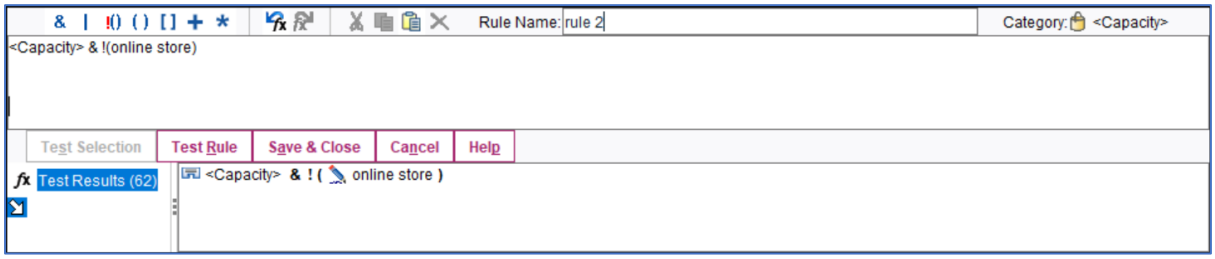


Figure 7.23 The completed expression in the rule editor

Once again, to return to the category pane, click:

Save & Close

7.4 Creating a Text Analytics Package

In the next section of this chapter, we will look at creating categories using automated methods. Before we do so, it makes sense for us to save our current categories in a separate file and start from scratch, so that it is clear which categories have been generated automatically and which ones have been created manually. If we wish to save our categorisation schema, we will also need to save the descriptors within the categories themselves, as well as the type groups that we created and all the other edited resources that affect the extraction process. Fortunately, there is a file type that allows us to save all of these elements in one place: the Text Analytics Package (or .tap) file.

Text analytics packages, or TAP files, are a useful way to save entire projects in a single file. Furthermore, TAP files allow the user to work with multiple sets of categories that refer to different topics or separate stages or iterations of a project. When working with TAP files, users can control how they update their work. They can make improvements to existing category sets, edit linguistic resources, or create new category sets before updating the existing TAP file. The update procedure allows users the choice of appending category sets, replacing resources, editing the package label, and renaming/reordering existing sets. Figure 7.24 shows the structure of a TAP file.

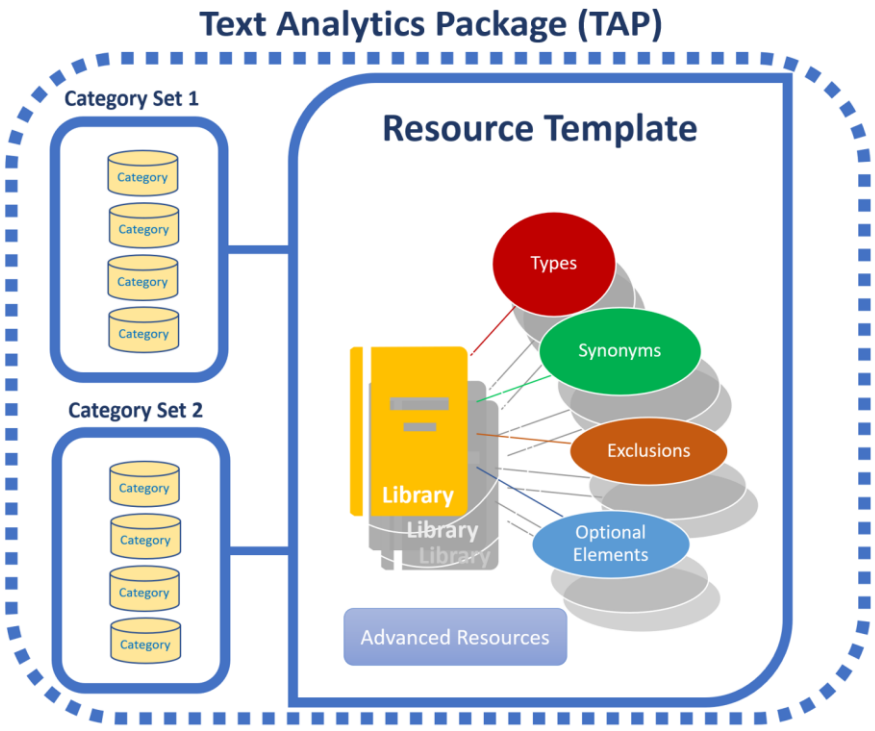


Figure 7.24 The structure/contents of a Text Analytics Package (TAP) file

To save our current categories as part of a TAP file, from the main menu click:

File

Text Analytics Package

Make Package

Figure 7.25 shows this procedure.

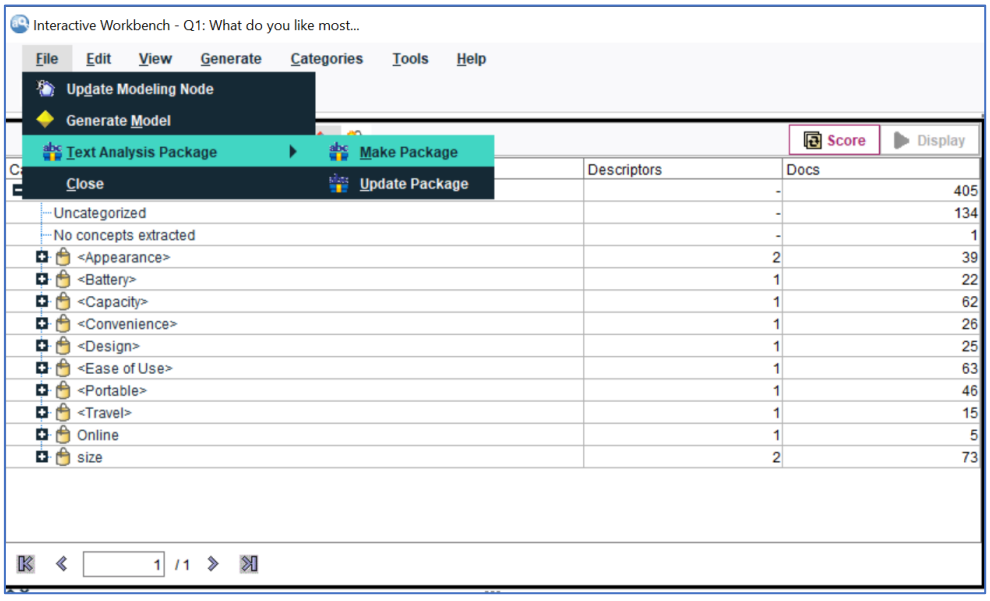


Figure 7.25 Creating a Text Analytics Package (TAP) file

This opens the **Make Package** dialog. Here we can specify a file name, package label and name for the category set. Specify the file name:

MP3_Survey

Within the set label box, replace the existing label with:

Manual Categories

Figure 7.26 shows this procedure.

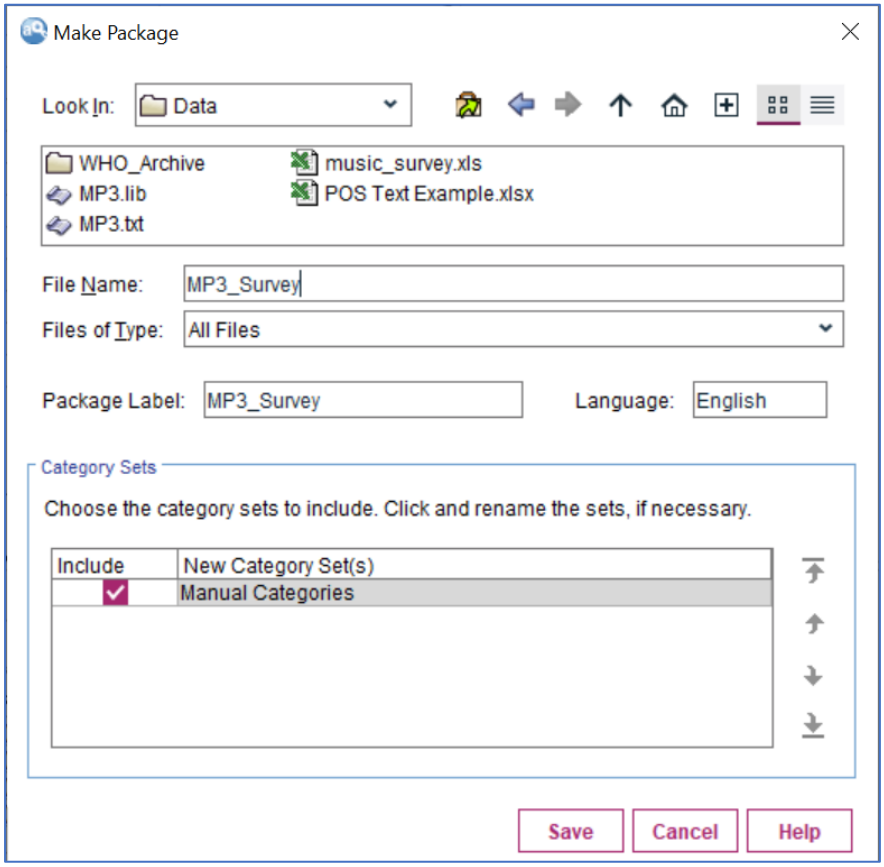


Figure 7.26 Making a Text Analytics Package (TAP)file

Now click:

Save

Finally, select all the existing categories in the category pane and press:

Delete

Practice Exercise – Chapter 7

Within the folder **Student Exercises** open the following stream:

Chapter_07_Practice.str

1. Right-click on the text mining node and change the loaded resource template from **Basic Resources (English)** to **Car Rental**. Now, run the text mining node.

2. Once the extraction has completed, we can begin the process of manually creating categories. We can begin by selecting some individual concepts. Right-click on the concept **friendly** and select **Add to Category** followed by **Create New Category**. You will notice that the category is immediately created in the **Category** pane. Follow the same procedure using the following concepts to create new categories:

- no problem
- helpful
- available
- courteous

Right-click on the category **available** and rename it to **availability**.

Find the concepts **knowledgeable** and **willing to help** and right-click on them to add them to the category **helpful** or drag and drop them each directly into the category.

3. Now switch the extraction pane from the **Concept** view to the **Type** view. Create new individual categories from the following types:

- <WaitTime>
- <Upgrade>
- <Air Travel>
- <Pick_Up_Drop_Off>
- <Fast>

Add the type <**slow**> to the category <**WaitTime**>.

At this stage, more than half of the cases have matched with at least one of the categories.

4. We can clean up the appearance of the categories themselves by merging them or renaming them. Collapse the categories and rename them in the following way.

Old Category Name	New Category Name
friendly	Friendly
no problem	No Problems
helpful	Helpful
available	Availability
courteous	Courteous
<WaitTime>	Wait Times / Slowness
<Upgrade>	Upgrades
<Air Travel>	Air Travel
<Pick_Up_Drop_Off>	Pick Up / Drop Off
<Fast>	Quickness

5. Right-click in the category pane, select **Create Empty Category**. Give the empty category the name **Reservations**. Right-click on the newly created category and click **Create Category Rule**. Name the rule **Reserve_Rule**. Type the following syntax in the rule editor:

```
reserv*
```

Click the button **Test rule**. It should match about 4 cases. Click **Save & Close** to return to the Category pane. Score the data by clicking **Score**.

6. If there is time, feel free to create any additional categories as you see fit.

7. From the **File** menu, click **Text Analysis Package** and choose **Make Package**. Navigate to the **data** folder in **C:\SV Training\Text Mining\Data** and give the text analysis package the file name **Car_Rental**.

Before you save the file, in the box labelled **New Category Set(s)** assign the label **Manual Categories.**

Now click **Save**.

8. Exit the interactive workbench without updating the node.

Chapter 8 Automatic Categorisation

Modeler Text Analytics contains two primary methods for the automatic generation of categories. The default method is based on linguistic algorithms. An alternative method enables users to build categories using frequency-based techniques. The automatic categorisation options are potentially extremely useful, not least because:

- They can be combined with manual categorisation.
- They may identify new topics or categories that the user did not consider.
- They may identify alternative ways to categorise identified topics.
- They may find stray concepts that the user had not considered when creating categories manually.

8.1 Automatic categorisation with default settings

As we begin this section with no defined categories, we can explore the effect of running automatic categorisation with the just default settings. To do so, from the main menu click:

Categories

Build Categories

Build Now

Immediately we can see that more than 20 categories are generated by the system (see Figure 8.1).

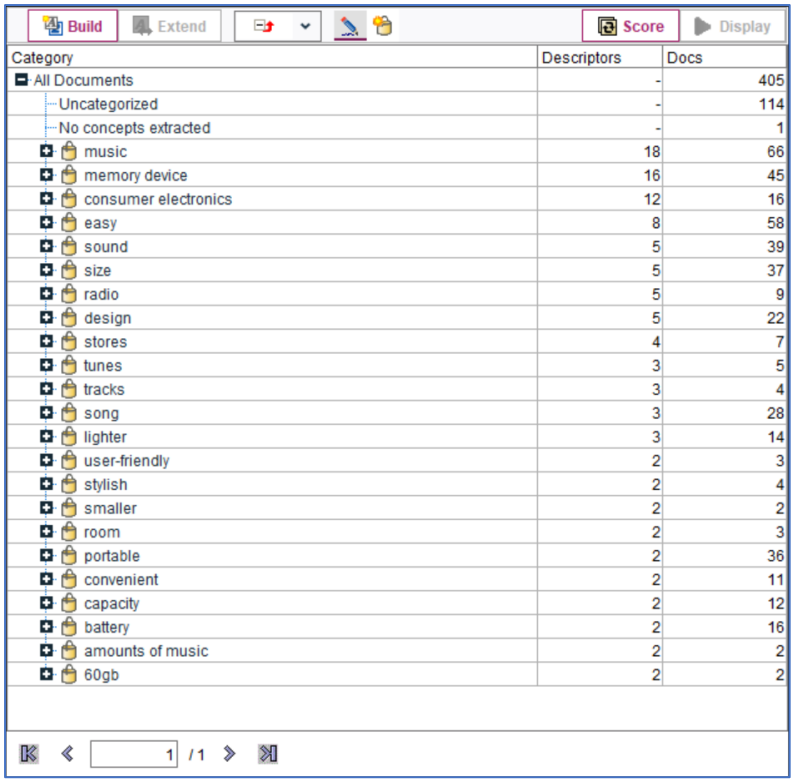
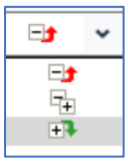


Figure 8.1 Categories automatically generated with the default settings

Among the first things that we may notice are:

- Many of the categories overlap with one another thematically (e.g., **music**, **tunes**, **tracks** and **song**)
- Some of the categories have more than 10 descriptors (**music**, **memory device** and **consumer electronics**)
- Some of the categories only contain 2 records (e.g., **smaller**, **amounts of music** and **60gb**)
- Many of the categories are similar to the ones we created manually (e.g., **portable**, **capacity** and **design**)

If we click the Expand All button, we'll be able to see how each category is composed. Within the categories pane, click:



As Figure 8.2 shows, we can immediately see that the categories are almost entirely composed of compound concepts (there are no type groups). There is a single instance of an automatically generated rule within the **song** category. Moreover, the colour-coding indicates, that the various concepts within each category all belong to

the same type group. Lastly, we can see that the default automatic categorisation has created a number of subcategories within some of the category groupings.

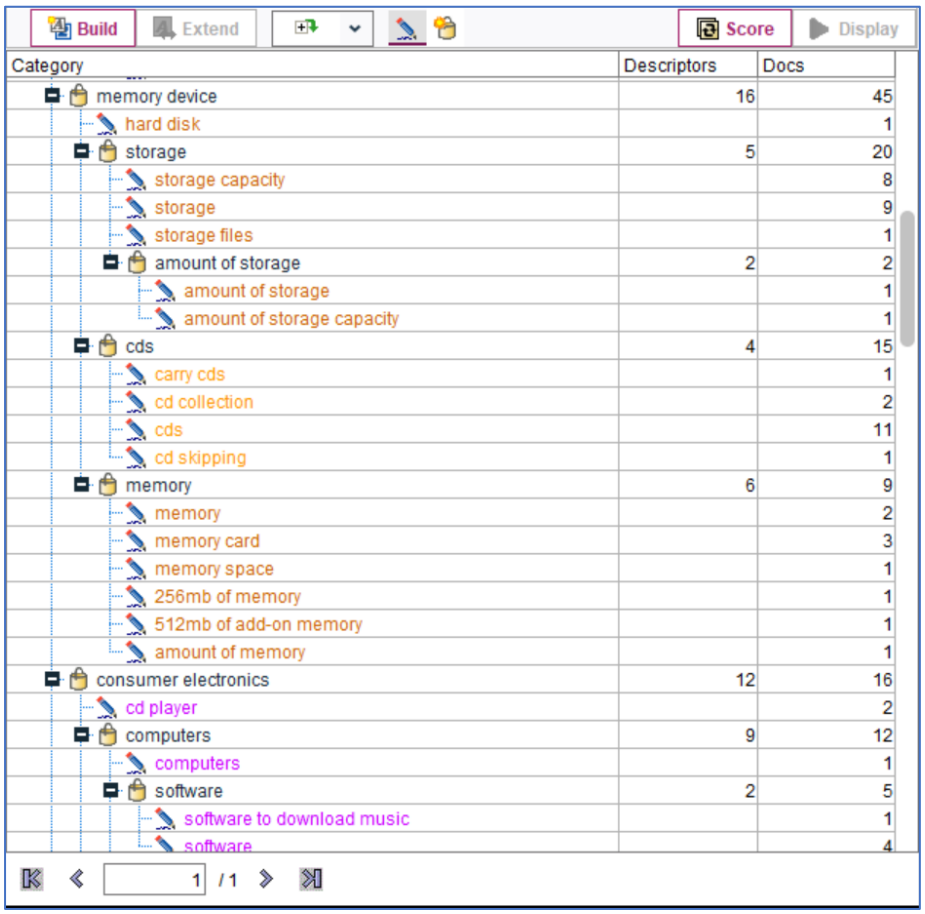


Figure 8.2 Expanded categories revealing descriptors and subcategories

At this stage, the user has a number of options they might want to consider, such as:

- Merge several categories (and/or their subcategories) like **60gb**, **memory device** and **capacity** on the basis that they refer to the same topic and will result in a simplified categorisation schema.
- Delete redundant categories like **sound** or **music** on the basis that they are too generic to be regarded as insightful.
- Flatten all the categories to create a simpler overall structure.

Having cleaned up the results, the user could then begin the process of enhancing the categorisation with their own categories and adding concepts and types to the existing categories. In this way, automatic categorisation might act as a jumping off point for the entire categorisation process.

8.1.1 Automatic categorisation settings

Before we look at the various custom options associated with the automatic categorisation engine, once again:

Select and delete all the existing categories

From the main menu click:

Categories

Build Settings

The build settings dialog appears. It shows that the categories will be built from the descriptors within a list of pre-selected types. By default, only the types that capture the most records are pre-selected. Also, by default types from the Opinions and Budget libraries are deselected. If there are interesting type groups here that you feel should be included in the automatic categorisation, then by all means select them.

You can also see that there are two radio buttons in the **Techniques** section of the dialog that allow the user to choose between using **linguistic techniques** or **frequencies** to build categories. The default selection is linguistic techniques. Figure 8.3 shows this dialog.

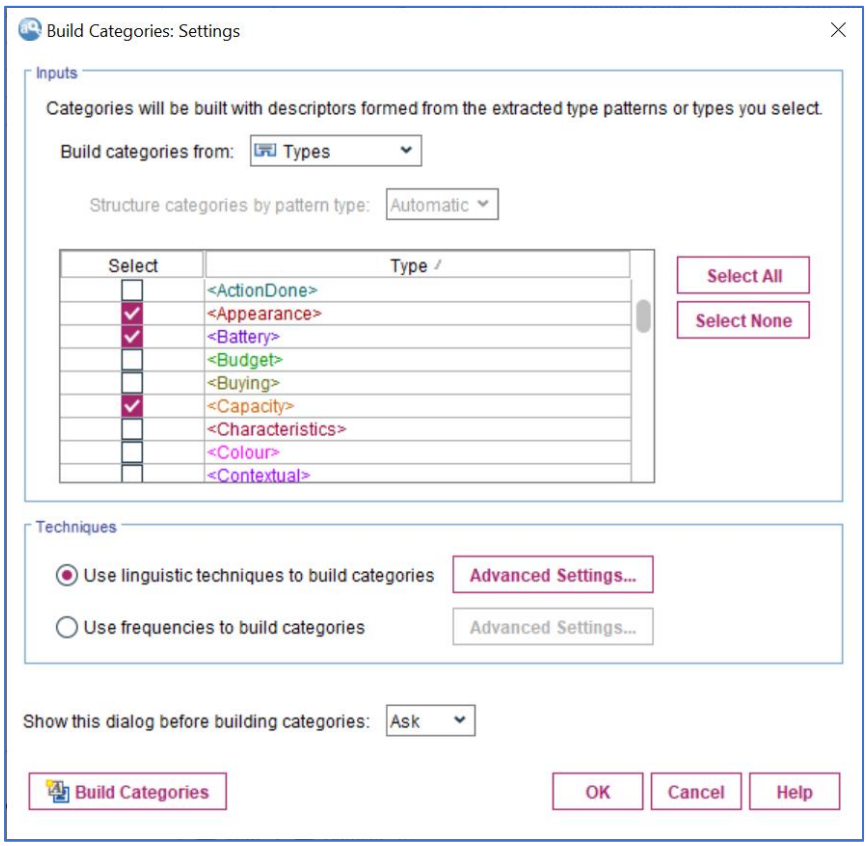


Figure 8.3 The Build Categories: Settings dialog

To view the advanced settings associated with the linguistic techniques option, click the button marked:

Advanced Settings

The advanced settings linguistics dialog is generated. This contains several options and settings that control how the linguistics-based automatic categorisation occurs. Figure 8.4 shows a numbered annotated image of this dialog.

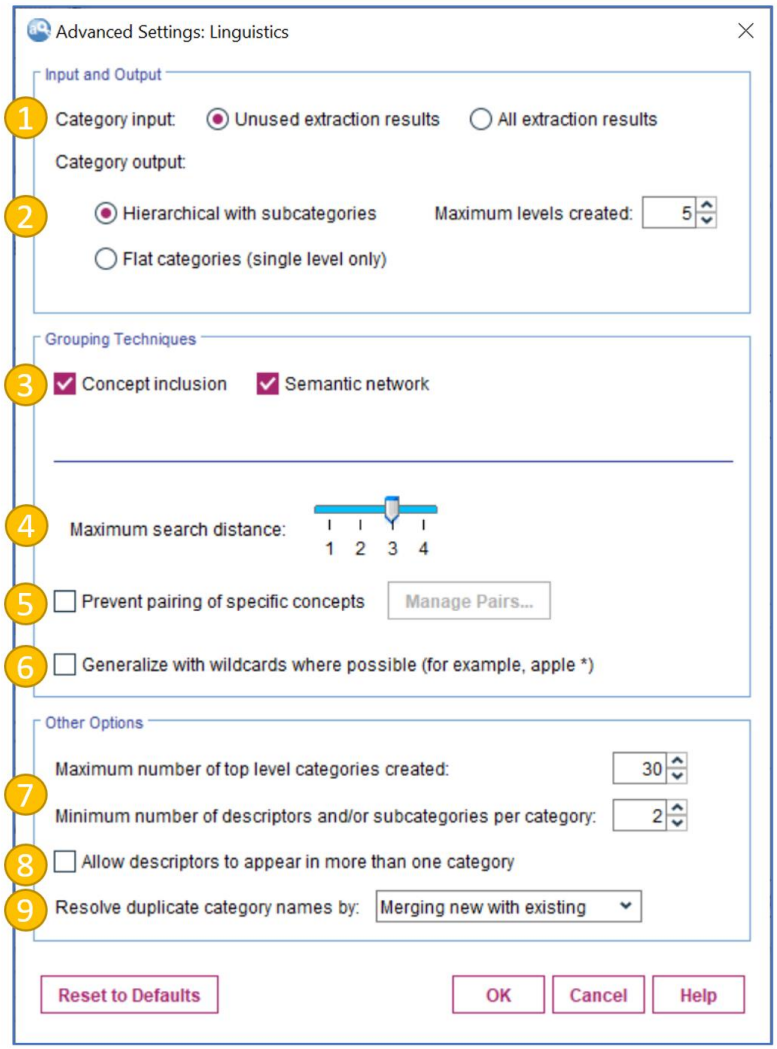


Figure 8.4 Annotated image of the Advanced Settings: Linguistics dialog

The numbered annotations within Figure 8.4 refer to the following options:

1. **Category Input:** These options control which extraction results are to be used in the auto-generated categories. If no categories have yet been created, then the results are identical.
    - **Unused extraction results:** This mode uses extraction results that haven't already been used as descriptors in any existing categories. This method reduces the likelihood that categories will overlap with one

another. The effect to is to produce fewer categories with fewer cases matching with multiple categories. This is a useful option when one wishes to avoid changing the existing categorisation schema too much.
    - **All extraction results**: This option enables categories to be built using any of the extraction results, whether they are present in existing categories or not. This may result in more categories that may overlap with one another.
2. **Category Output:**
    - **Hierarchical with subcategories:** This is the default setting. It allows the creation of hierarchical categories. The **maximum levels** setting allows the user to control how many levels of subcategories can be created.
    - **Flat Categories**: This option prevents the creation of hierarchical categories. Flat categories contain no subcategories.
3. **Grouping Techniques:** This section of the dialog controls which categorisation algorithms are used.
    - **Concept Inclusion:** This method uses an algorithm that creates categories by identifying a root term and finding other concepts that include it in phrases. For example, the term **training** may appear as part of a compound phrase such as **staff training** or **training course**. Word order is ignored by this method, so even root terms that have multiple words such as **text mining course** can be grouped with phrases like **course in text analytics**. It's an approach that can work well with domain-specific terms that the system may not recognise.
    - **Semantic Network:** This method generates categories using a semantic/lexical network based on WordNet®, a linguistic project based at Princeton University. WordNet® is a large lexical database of nouns, verbs, adjectives and adverbs that have been grouped into sets of synonyms. However, the database also arranges the words into categories. Specific words that belong to a more general category are known as *hyponyms*. As an example, a general category such as **fruit** can contain several hyponyms such as **apple**, **banana** or **grapes**. The sematic network algorithm's use of synonyms and hyponyms means that it can create categories containing terms that are similar to each other or belong to the same class of object. The semantic network tends to produce fewer categories than concept inclusion and performs less well with technical or jargonistic terms.
4. **Maximum search distance:** This option only applies to the semantic network method. It is controlled by a slider switch which ranges from 1 to 4. The selected value refers to how far the technique is allowed to search in order to create categories. Lower values force the categorisation to be more restrictive, resulting in less ambiguous also fewer categories containing fewer records. Higher values allow the semantic network to create a larger number of more populous, but looser categories.

5. **Prevent pairing of specific concepts:** This checkbox enables a useful option that stops the process from erroneously grouping pairs of concepts (e.g., **hot dog** and **dog**).
6. **Generalize with wildcards where possible:** This option allows the system to generate generic rules in categories using the asterisk wildcard. So instead of a category containing multiple rules with separate descriptors such as **training course**, **training requirement** and **training attendee**, the rule might simply contain the term **training ***. It's possible that one may end up with the same number of records matching with the category whether this option is used or not, but wild cards tend to reduce the number of descriptor rules in a category and therefore simplify it.
7. **Maximum Categories and Minimum Descriptors/Subcategories:**
    - **Maximum number of top-level categories created:** This option allows the user to limit the number of categories generated when using the automatic classification techniques. It might be worth experimenting with this setting by choosing a high value initially and then deleting any of the less useful categories.
    - **Minimum number of descriptors and/or subcategories per category:** This option specifies the minimum number of descriptors and subcategories a category must contain in order to be created. Editing this option may help to prevent creating categories that contain too few records.
8. **Allow descriptors to appear in more than one category**: Concepts can naturally belong to more than one category. Switching on this option allows a concept like **text mining course** to be automatically added to the categories **text mining** and **course**. If this option is not selected, the concept will be added as a descriptor to only one of these categories. In which case, there are a number of factors that determine which category would be chosen including the number of records in which **text mining** and **course** occur. The alternative approach would be to ensure that the terms **text mining** and **course** are extracted as separate concepts, for example through the use of type groups.
9. **Resolve duplicate category names by**: This drop-down menu contains two options controlling how the system handles situations when automatic category generation creates a category that already exists. The default option, **Merging new with existing**, simply means that categories with the same name would be merged. The alternative option, **Skip/do not create duplicate**, prevents the creation of a category if one with the same name already exists.

8.2 Automatic categorisation with customised settings

As mentioned earlier, perhaps the most useful aspect of automatic categorisation is the fact that it can save so much time compared to manual methods. That being said, here are some points of advice when using this approach:

- Spend time experimenting with different settings to find an optimal set of results.
- Don’t be afraid to delete many of the categories that this method generates.
- Consider renaming categories that contain useful descriptors but are poorly described.
- Consider merging categories if appropriate.
- Think about whether or not you want hierarchical categories.
- Pay particular attention to useful categories that you might not have created using manual methods.
- Pay attention to the type groups in the **Build Categories: Settings** dialog. Are there any unchecked ones that you would like to include?

8.2.1 Semantic network with all types

We’ve already seen the results from using just the default settings and running the **Build Categories** command. Let’s look at the effect of changing some of the default options and creating a customised categorisation. From the main menu, click:

Categories

Build Settings

Within the initial **Build Categories: Settings** dialog, to select all the type groups to be included in the process, click the button marked:

Select All

Now click:

Advanced Settings

Within the **Advanced Settings: Linguistics** sub-dialog, in the **Grouping Techniques** area, uncheck the box marked:

Concept Inclusion

Now only the Semantic Network grouping method will be used. Figure 8.5 shows both of these dialogs.

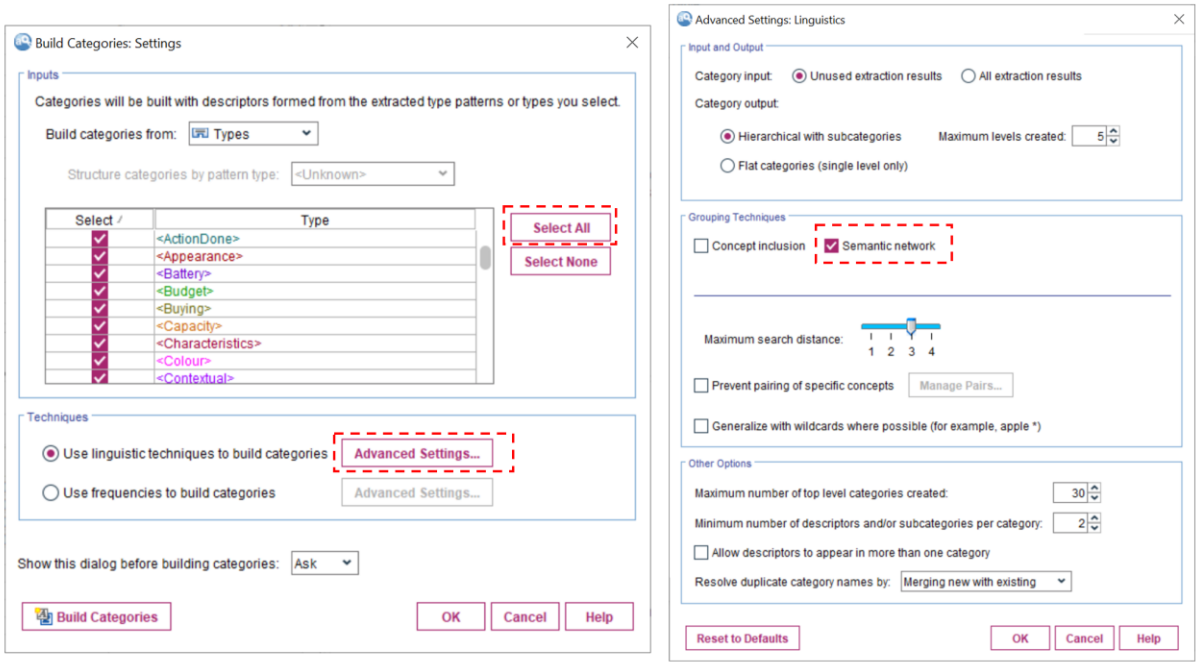


Figure 8.5 Choosing Semantic Network with all types for automatic categorisation.

To continue click:

OK

Build Categories

The categories are now generated automatically. Remember that Semantic Network attempts to generate categories based on sets of synonyms within a network of hyponyms. This method tends to be quite hit and miss and is prone to creating some strange categories. For example, our most populous category is named **music acoustics and physics**. It contains 33 records and uses 3 descriptors. Of these 33 records, 31 are associated with the term **excellent** which has been used to form a subcategory with the term **heavy metal**. Can you guess why? Clearly this category is not about music or musical genres per se. There are also strange categories called **england** and **philippines** that have little to do with those geographical regions. We can also see that 116 records remain uncategorised. Nevertheless, this method *has* managed to create sensible categories such as **memory device**, **finance** and **computer network**, which perhaps only need to be renamed in order to be useable. To continue our exploration of different automatic categorisation approaches once again:

Select all the categories and delete them

8.2.2 Concept inclusion with flat categories and wildcard generalization

In this iteration, we will once again include all the type groups to build categories from. However, on this occasion we will give the system even more freedom to create categories. Figure 8.6 shows the changes we will make to the advanced settings options:

Flat categories

Concept inclusion only

Generalize with wildcards where possible

Maximum number of top level categories created: 45

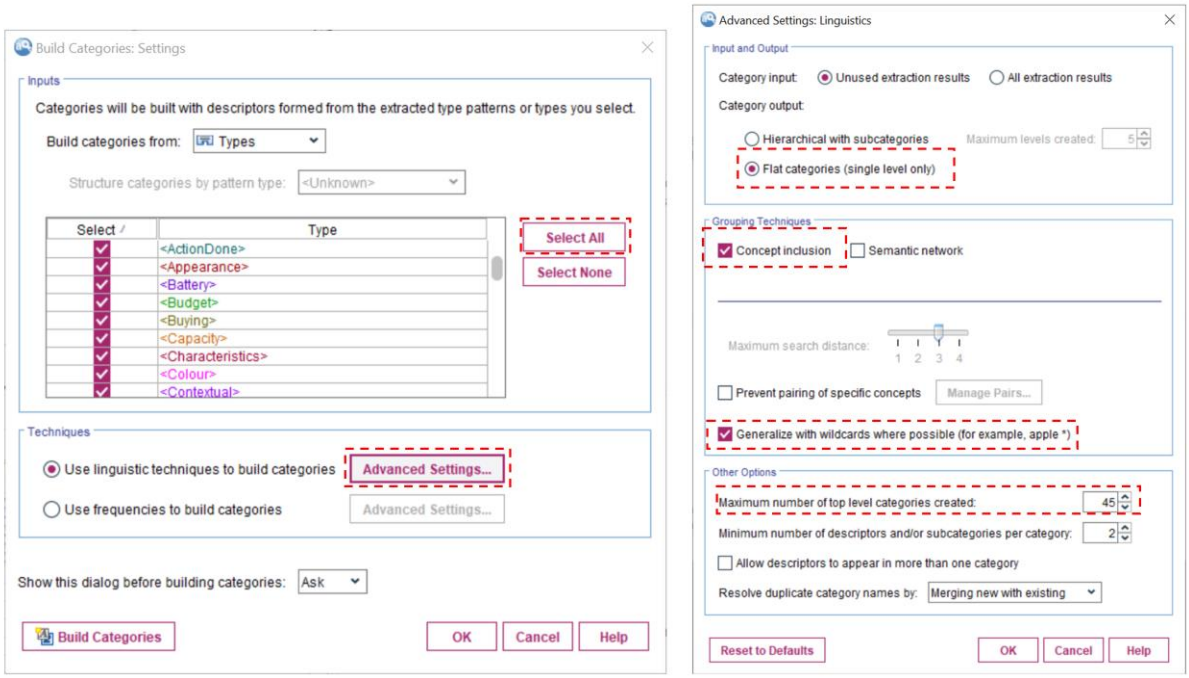


Figure 8.6 Choosing Concept inclusion, flat categories and wild card generalization with all types

When we re-run the categorisation, the results are quite different from the previous iteration. Firstly, we have a lot more categories (45 in total) and there are only 92 uncategorised cases. Secondly, we can see that the most populous categories contain simple wild card rules such as **easy *****. We can also see that there are lots of categories that could be merged as they refer to different aspects of the same thing such as **capacity**, **storage**, **stores** and **amounts of music**. If we use the **display** button in the category pane to show the *uncategorised* records, we can see that there are a number of responses associated with the size of the product and its appearance that the automatic categorisation has ignored. Figure 8.7 shows some of these records.

		Q1: What do you like most about this portable music player? (92)	Categories
19	45.0	...Everything! Product A rules! I can't wait to get a video one!...	
20	221.0	...Makes me feel young again!...	
21	49.0	...It has some DJ features like building track list, cross fading, etc. It's very durable...	
22	233.0	...It's small and light...	
23	164.0	...It was much more affordable than Product A....	
24	310.0	...the colour of the device...	
25	2.0	...The battery power is great....	
26	152.0	...its small...	
27	187.0	...It's way cool, the teacher can't even see the headphone cable....	
28	215.0	...I was given it as a present, so I just like it....	
29	1.0	...little, light...	
30	219.0	...It fits comfortably in the palm of my hand...	
31	292.0	...its color...	
32	5.0	...The shuffle mode....	

Figure 8.7 Uncategorised records from concept inclusion with flat categories and wildcard generalization

Generally speaking, this iteration did a good job of finding the majority of topics and characteristics associated with the MP3 product, though there might still be some work to clean up the categories and to utilise some of the unused extracted concepts.

8.2.3 Concept inclusion with a maximum of 20 flat categories and minimum 5 descriptors

An alternative approach is to use the linguistic automatic categorisation methods to create a limited number of the most prevalent categories and then build up from there. To explore this, we can return to the **Build Categories: Settings** dialog and the **Advanced Settings: Linguistics** sub-dialog and make the following changes (shown in Figure 8.8):

Uncheck the* Generalize with wildcards where possible *option

Maximum number of top level categories created: 20

Minimum number of descriptors and/or subcategories per category: 5

Users should be aware that allowing wildcard generalization effectively switches off the minimum descriptors setting, as a descriptor with a wildcard can refer to any number of compound concepts.

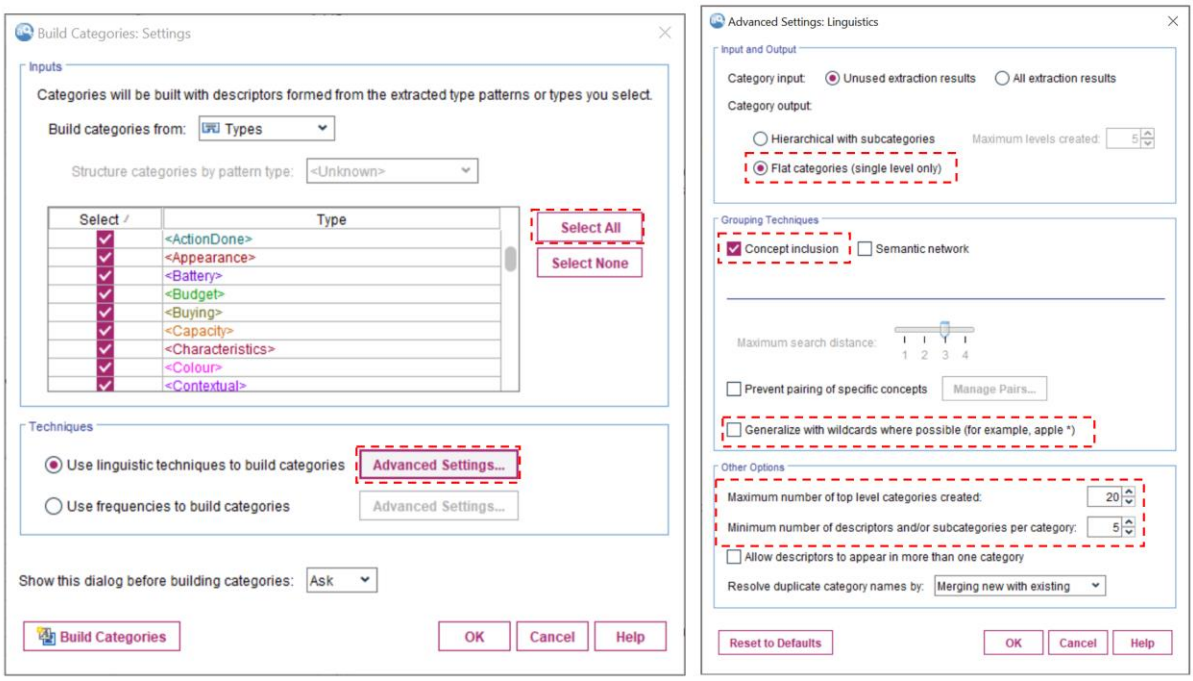


Figure 8.8 Choosing Concept inclusion, flat categories with maximum categories = 20 and minimum descriptors = 5

Now when we run the Build Categories procedure, we see only 7 categories created. Indeed, the category with the smallest record frequency, **feature**, isn't very useful and arguably, should be deleted.

8.3 Extending categories

An interesting aspect of the categorisation process is the ability to extend categories. Extending refers to a process which automatically adds or enhances the descriptors in order to capture more data within existing categories. This broadens the category. Users can extend single categories, groups of categories or all categories. We can see the effect of this. Firstly, within the category pane:

***Delete the category* feature**

Shift-click to select all the remaining categories

From the main menu click:

Categories

Extend Categories

Extend Now

As Figure 8.9 shows, each category has the suffix **Extended** temporarily appended to its label.

Category	Descriptors	Docs
All Documents	-	405
Uncategorized	-	
No concepts extracted	-	
music Extended	1	
easy Extended	1	
size Extended	1	
storage Extended	1	
radio Extended	1	
memory Extended	1	

Figure 8.9 Category labels are temporarily appended with the suffix 'Extended'

Pressing the **Score** button and re-sorting the categories by their document count removes the **extended** label, and reveals that the categories **music**, **easy** and **size** have increased their share of the record count.

Category	Descriptors	Docs
All Documents	-	405
Uncategorized	-	233
No concepts extracted	-	1
music	18	65
easy	7	56
size	5	37
storage	5	20
radio	5	9
memory	6	9
feature	5	6

Figure 8.10 Categories and their respective document counts before being extended

Category	Descriptors	Docs
All Documents	-	405
Uncategorized	-	227
No concepts extracted	-	1
music	1	75
easy	1	62
size	1	38
storage	1	20
memory	1	9
radio	1	9

Figure 8.11 Categories and their respective document counts after the 'Extend' command is run

In Figure 8.10 we can see that the most populous category **music**, contains 18 separate descriptors. These descriptors are comprised of individual compound concepts containing the term **music**. When we investigate the same category after the **Extend** command has been run, we can see that these descriptors have now been replaced by a single descriptor wildcard rule: *** music ***. This rule will capture any mention of music within a compound concept and as such, it accounts for an additional 10 records.

In many ways, the **Extend** process is another version of the linguistic categorisation process we have been exploring thus far. So it is most effective when the categories

have been created manually, as it allows us to blend the two approaches. It also differs from the **Build Categories** procedure as it can be applied to categories on an *individual* basis. But just like that procedure, we can edit some of the options around how **Extend Categories** is performed. To view these options, from the main menu click:

Categories

Extend Settings

Figure 8.12 shows the resultant dialog.

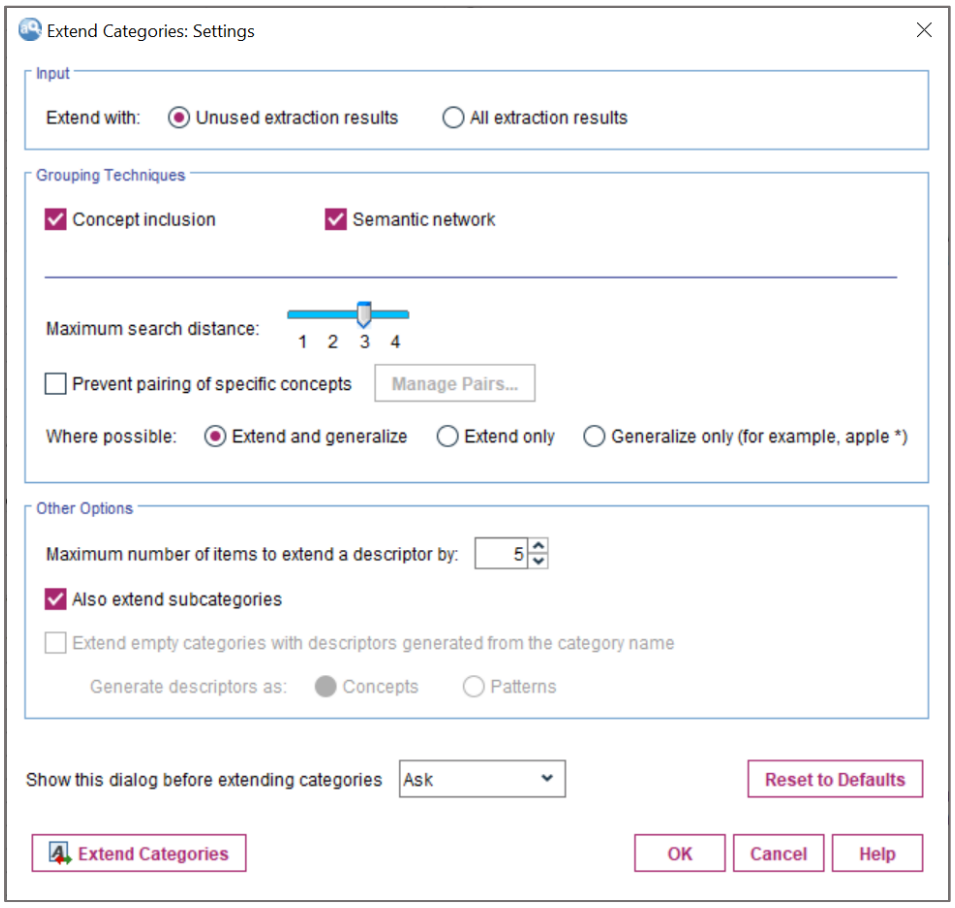


Figure 8.12 The Extend Categories: Settings dialog

Extend and generalize. This option will run the two linguistic grouping algorithms against the selected categories before using wildcards to generalize the descriptors. In our previous example, the categories had already been automatically generated using the linguistics algorithms, so only the generalize option had an effect on the number of records categorised.

Extend only. This option only uses the linguistic grouping algorithms without wildcard generalization. It may be useful to run the **Extend only** option for manually created categories before running the **Extend and generalize** option to see the incremental effects of both approaches.

Generalize only. This option simply generalizes the descriptors without running the linguistic algorithms.

It's worth bearing in mind that extending categories has no effect on categories comprised only of descriptors made from type groups like those we created in the previous chapter. However, as we will see, the procedure does allow empty categories to be populated based on the category name. This can be useful if we wish to import a file that already contains category structures such as a coding frame.

8.4 Frequency-based categorisation

So far, we have investigated manual categorisation, automatic linguistics-based categorisation and techniques for extending existing categories. You may recall that the **Build Categories** procedure also allows us to create categories based on the frequency. To illustrate this:

Delete any existing categories in the Categories pane

From the main menu, click:

Categories

Build Categories

Edit

Once again, the **Build Categories: Settings** dialog is displayed. This time however, we will choose the alternative primary mode for creating categories. Choose the option:

Use frequencies to build categories

On this occasion we will leave the default selection of type groups as they are. Figure 8.13 shows the dialog at this stage.

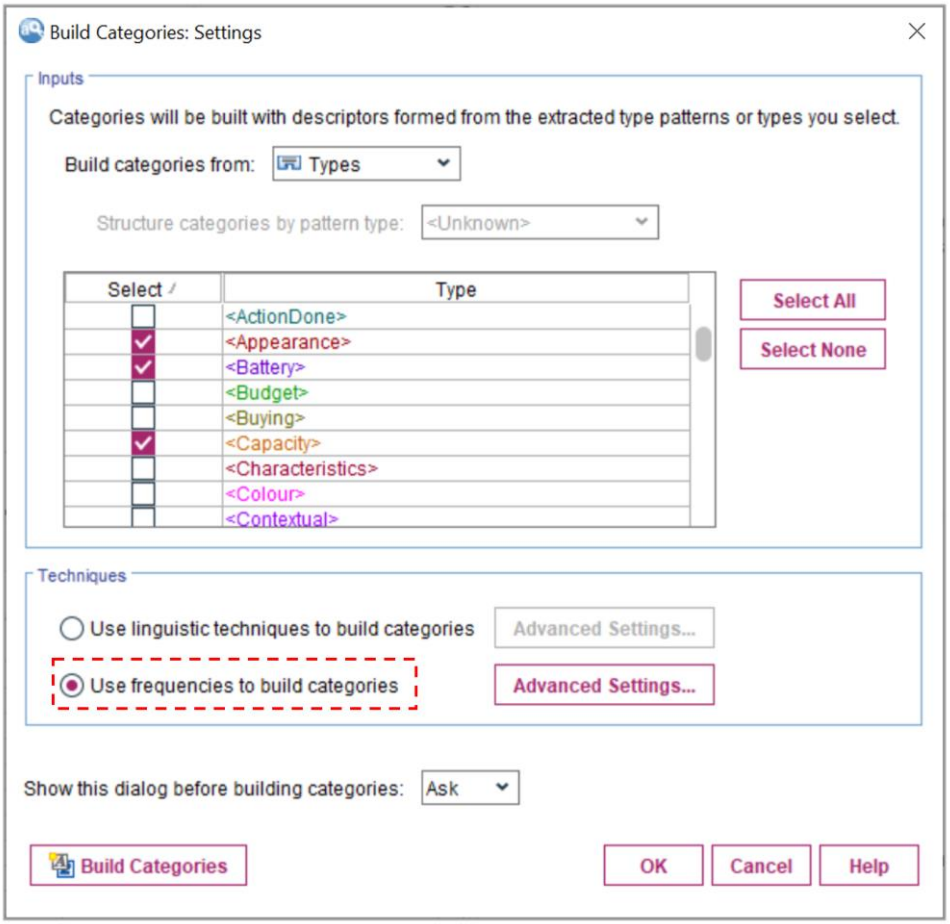


Figure 8.13 The Build Categories: Settings dialog with the 'Use frequencies to build categories' option selected

Before running the procedure, let's look at the options associated with this mode. To access the setting associated with frequency-based categorisation, click the adjacent button:

Advanced Settings

Figure 8.14 shows the **Advanced Settings: Frequencies** sub-dialog.

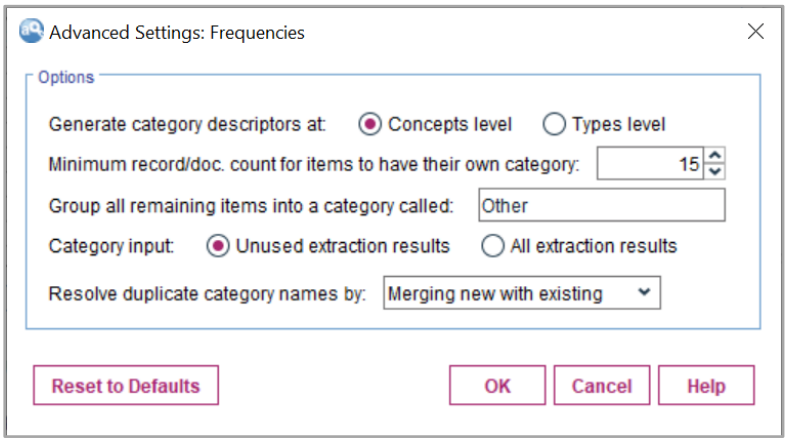


Figure 8.14 The Advanced Settings: Frequencies sub-dialog

The sub-dialog shows that in this mode, the procedure allows for categories to be generated based on descriptors at the concepts level or at the types level. The procedure also allows the user to control the minimum number of records that a category should contain. By default, all the remaining items are grouped in a category called **Other** (but this name can be edited). Once again, categories with duplicate names can be merged or not generated at all. To demonstrate the procedure, change the **Minimum record/doc. count for items to have their own category** to:

15

Then, to build the categories based on concept descriptors click:

OK

Build Categories

We can see that the process results in 17 categories with the **Other** category accounting for 193 records. The next most populous category is the **music** category which contains 48 records. We could have created more categories if we had chosen to select all the type groups for consideration in the **Build Categories: Settings** dialog. We can see that apart from the **Other** category, all the categories contain a single descriptor each in the form of an extracted concept. If we delete the **Other** category, there remains 129 uncategorised records. Despite its lack of sophistication, this method has nevertheless created quite sensible categories that closely match the manual procedures we investigated previously. This is partly because the diligent use of custom type groups has forced the extraction of many useful concepts. Figure 8.15 shows the category pane at this stage.

Category	Descriptors	Docs
All Documents	-	405
Uncategorized	-	129
No concepts extracted	-	1
music	1	48
music		48
small	1	47
small		47
easy to use	1	44
easy to use		44
portable	1	35
portable		35
sound	1	33
sound		33
size	1	27
size		27
song	1	26

Figure 8.15 Categories created automatically based on frequency of concepts note: category 'Other' is deleted

We can delete the categories and re-run the procedure, this time choosing to build frequency-based categories using types as descriptors. To so:

Select and delete all the existing categories

From the main menu click:

Categories

Build Categories

Edit

Advanced Settings

In the section marked **Generate category descriptors at**, choose the option**:**

Types level

Figure 8.16 shows the dialogs at this stage.

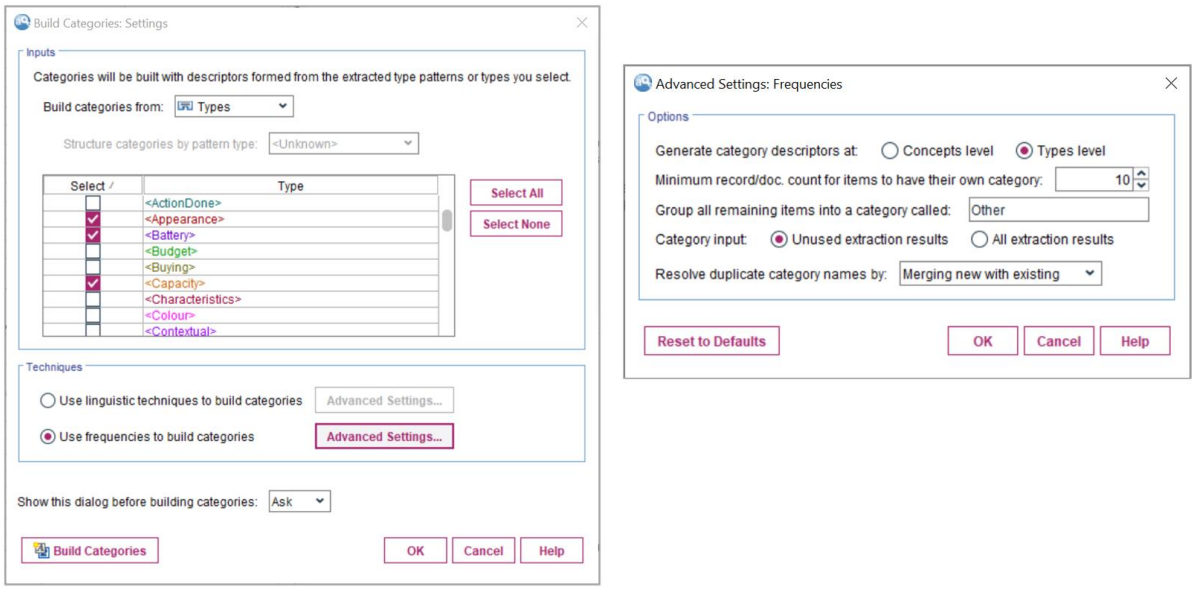


Figure 8.16 Using frequencies to build categories based on descriptors at the 'types level'

Click:

Ok

Build Categories

This time only 12 categories are created with no **Other** category generated. Interestingly, only 63 records remain uncategorised although this is partly due to the presence of the rather vague category **<Products>**. The most populous category is once again **<Music>** which accounts for 104 records. Again, this categorisation could form the basis for a more enriched and detailed final set of categories.

Figure 8.17 is a summary table showing the results of the different iterations of automatic categorisation that we have explored. If nothing else, it highlights the range of options afforded to users of the software when it comes to categorising the data with automated methods.

Method description	Additional types	Category count	Most frequent category/categories	Least frequent category/categories	Uncategorised records
Default	None (default)	23	music (65)	60gb, amounts of music, smaller (2)	116
Semantic Network	Select All	15	music acoustics and physics (33)	england, outdoor activities, plants, sports by types (2)	295
Concept inclusion, wildcards & category limit: 45	Select All	45	music (75)	tuning, surf, speakers, smaller, pc, organiser, amounts of music, 60gb (2)	92
Concept inclusion, category limit 45, descriptor limit 5	Select All	7	music (65)	feature (6)	233
Concept inclusion, category limit 45, descriptor limit 5 – Extended	Select All	6	music (75)	radio (9)	227
Frequency based, concept descriptors, minimum 10 records per category	None (default)	17	Other (193), music (48)	convenient, compact (10)	129 (without 'Other')
Frequency based, type descriptors, minimum 10 records per category	None (default)	12	<Music> (104)	<Ease of Use> (19)	63

Figure 8.17 Summary table showing the results of different approaches using the 'Build Categories' procedure in this chapter

8.5 Importing pre-defined categories

There may be circumstances where a set of categories has already been created as a **coding frame** for text responses. In these situations, it's possible to import the coding schema as an MS Excel file directly into the category pane.

To show how this is done, again, delete the existing categories, then from the main menu click:

Categories

Manage Categories

Import Predefined Categories

Figure 8.18 shows this process in action.

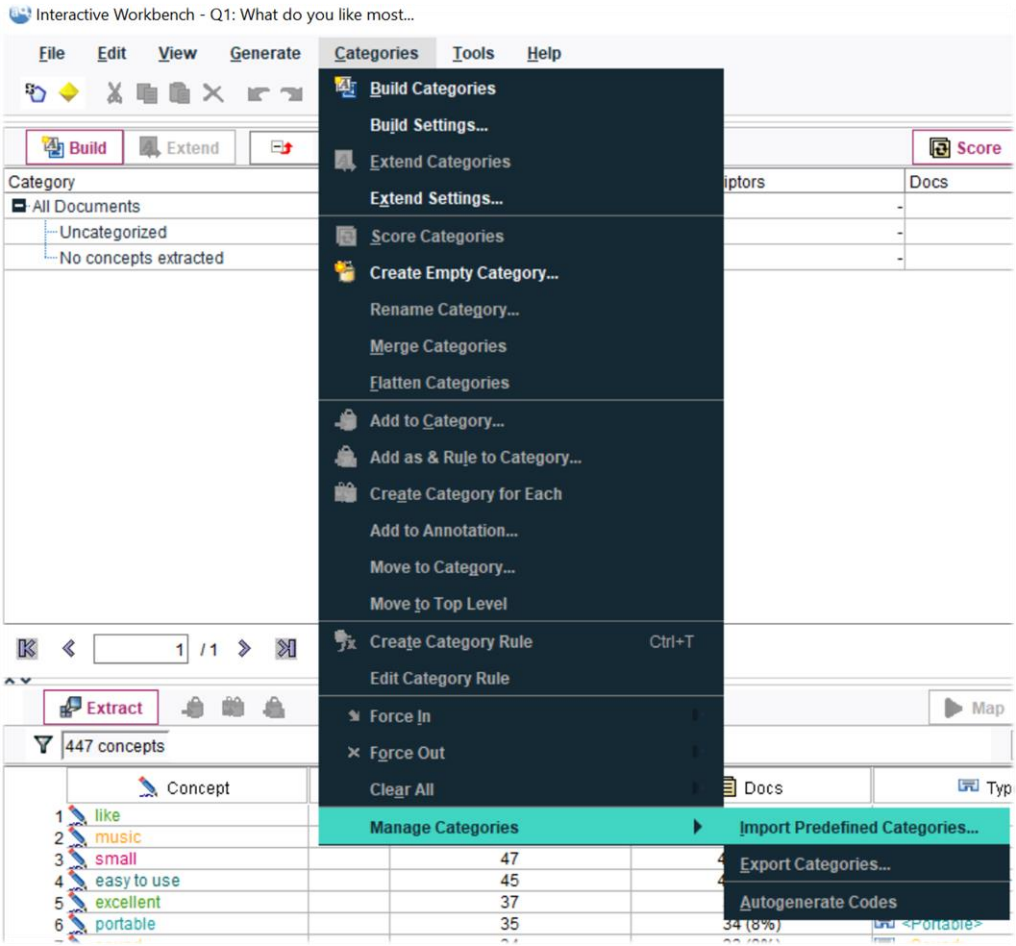


Figure 8.18 Importing Predefined Categories

This action generates the **Import Predefined Categories Wizard** (see Figure 8.19). To continue, from the **Data** folder choose the file:

MP3_Code_Frame.xls

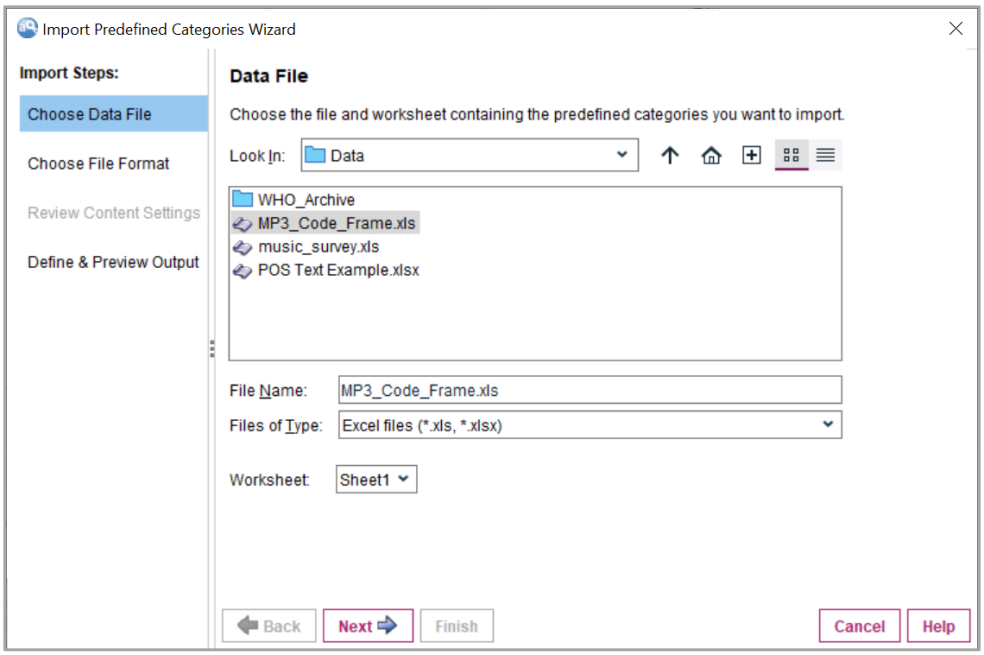


Figure 8.19 The 'Import Predefined Categories Wizard'

To continue the process, click:

Next

The wizard now advances to the **Choose File Format dialog**. We can see from Figure 8.20 that this allows us to import code frames with different category structures such as hierarchical indented formats. The default option in the wizard is to autodetect the format but in this case, we will explicitly choose the following option:

Flat list format: no subcategories

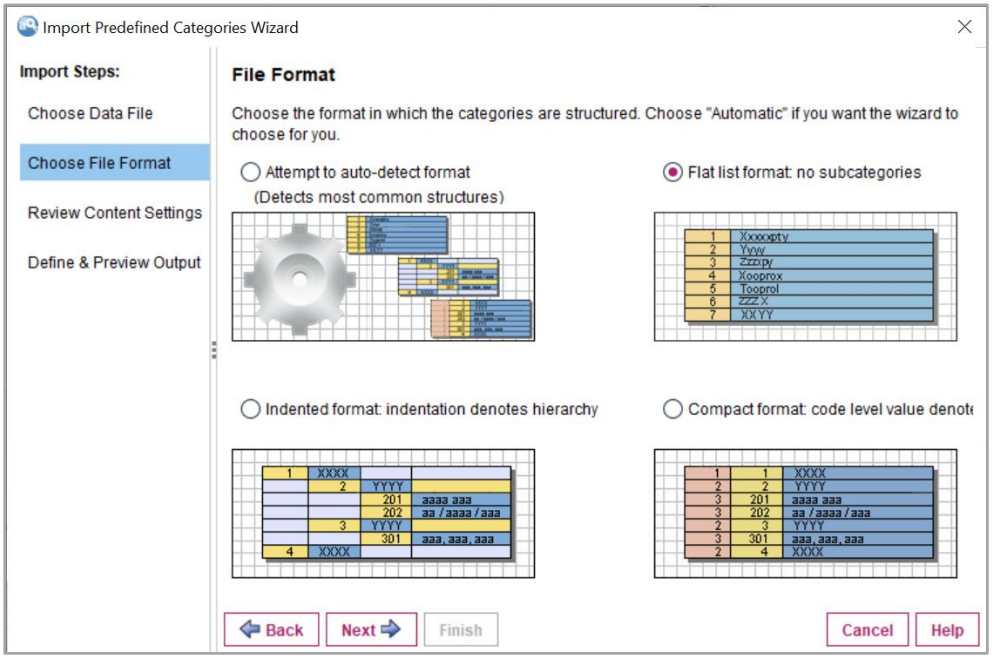


Figure 8.20 Choosing a category structure format

Now click:

Next

As Figure 8.21 shows, the wizard now advances to the **Content Settings** dialog where the user can review the contents of the file itself. In our example, the categories also contain various descriptors and a date stamp which the system automatically detects and colour codes. This particular sample file contains descriptors, as it was previously exported as a code frame directly from Modeler Text Analytics. However, it could just as easily have been created manually or by a third-party system.

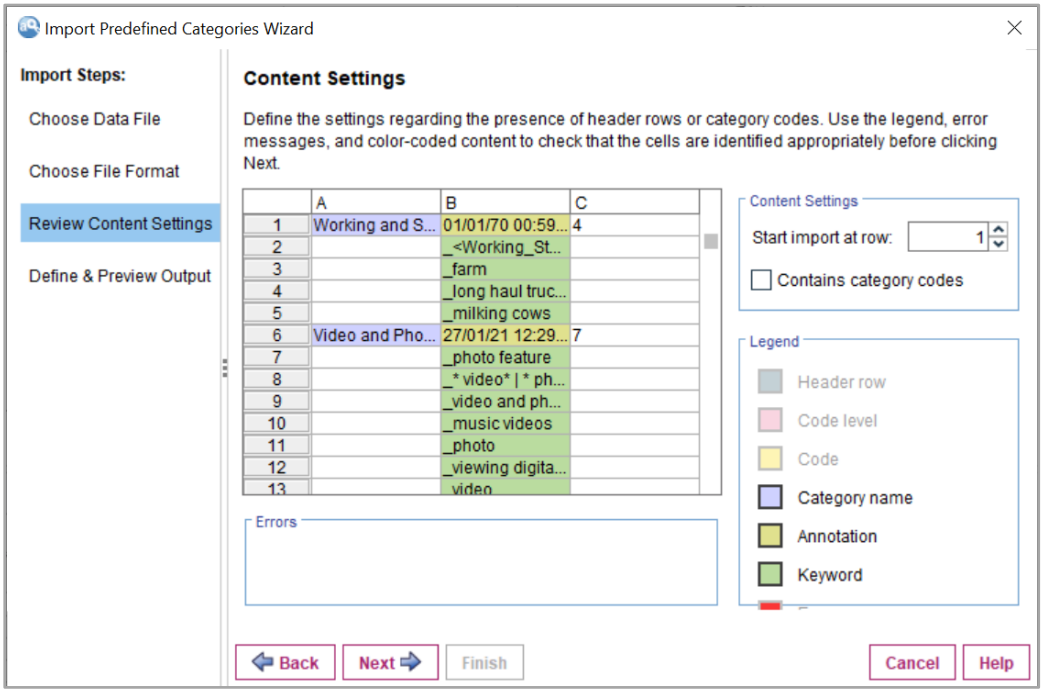


Figure 8.21 The 'Content Settings' dialog in the 'Import Predefined Categories Wizard'

To continue to the final stage of the wizard, click:

Next

Now we reach the **Define & Preview Output** stage of the wizard. Note that this offers the option to replace or merge the imported categories with any existing categories in the category pane. You may also notice that the procedure will import any keywords as descriptors to populate these categories and also to **Extend categories by deriving descriptors**. This useful option will generate descriptors from the names of the categories as well as any subcategories or annotations. To make use of this option check the box marked:

Extend categories by deriving descriptors

Figure 8.22 shows the dialog at this stage.

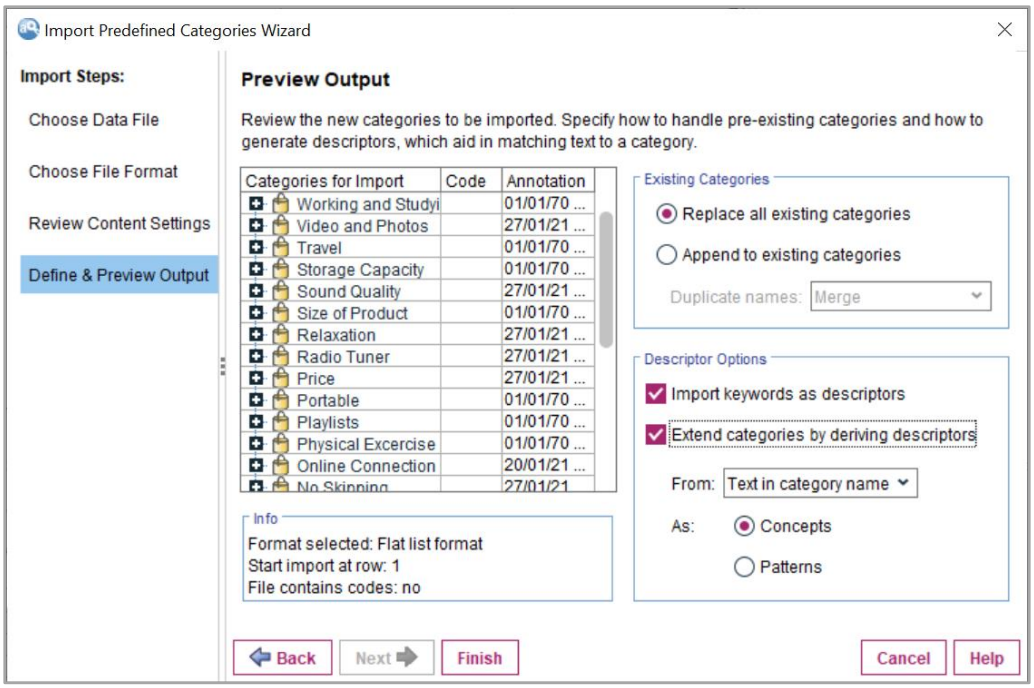


Figure 8.22 The final stage of the 'Import Predefined Categories Wizard': the 'Define & Preview Output' dialog

To complete the process, click:

Finish

The categories are imported, and the descriptors are parsed by the system. Figure 8.23 shows the results of this process. Note that several categories have the temporary **Extended** label attached showing that the **Extend categories** option has been applied to derive additional descriptors.

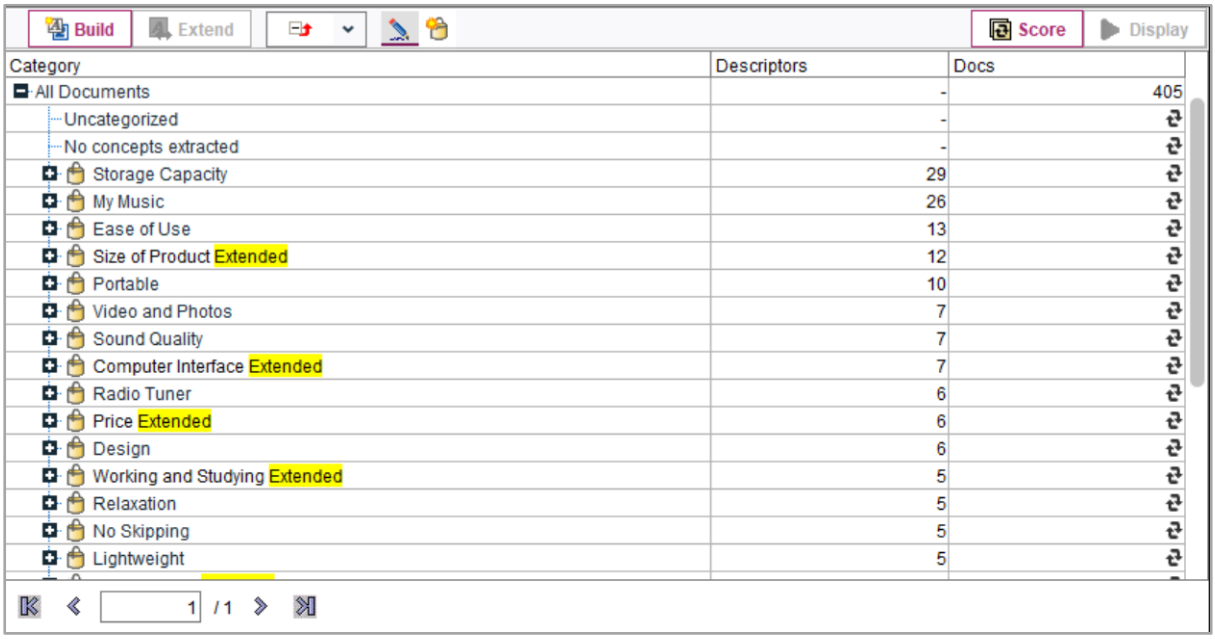


Figure 8.23 Categories imported from an external file

8.6 Visualising categories

The fourth pane in the categories and concepts window allows the user to create some basic charts that illustrate the relative frequencies of the categories as well as their interrelationships.

To use these charts, we first need to score the data and populate the categories, within the category pane. To do so, click the button marked:

Score

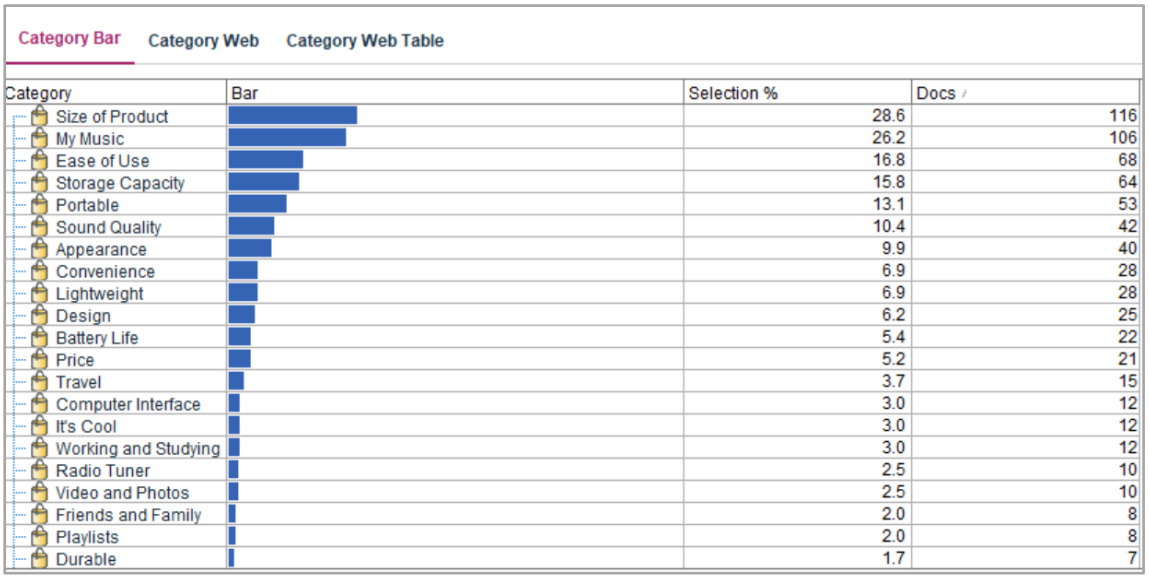
Within the categories pane, click:

All Documents

Now click:

Display

Within the adjacent visualisation pane, a bar chart showing a percentage and frequency breakdown of the categories is displayed in the **Category Bar** tab. Figure 8.24 shows the categories sorted in descending order of frequency.



Category	Bar	Selection %	Docs
Size of Product		28.6	116
My Music		26.2	106
Ease of Use		16.8	68
Storage Capacity		15.8	64
Portable		13.1	53
Sound Quality		10.4	42
Appearance		9.9	40
Convenience		6.9	28
Lightweight		6.9	28
Design		6.2	25
Battery Life		5.4	22
Price		5.2	21
Travel		3.7	15
Computer Interface		3.0	12
It's Cool		3.0	12
Working and Studying		3.0	12
Radio Tuner		2.5	10
Video and Photos		2.5	10
Friends and Family		2.0	8
Playlists		2.0	8
Durable		1.7	7

Figure 8.24 Category Bar chart in the visualisation pane

To further explore the visualisation options, click the tab marked:

Category Web

A web plot showing the how the various categories are interconnected is now shown. To change web plot to circular display, from the toolbar menu above the web plot click the following button:

To control which connections are displayed, click the following button to reveal a slider control:

The web plot at this stage is shown in Figure 8.25.

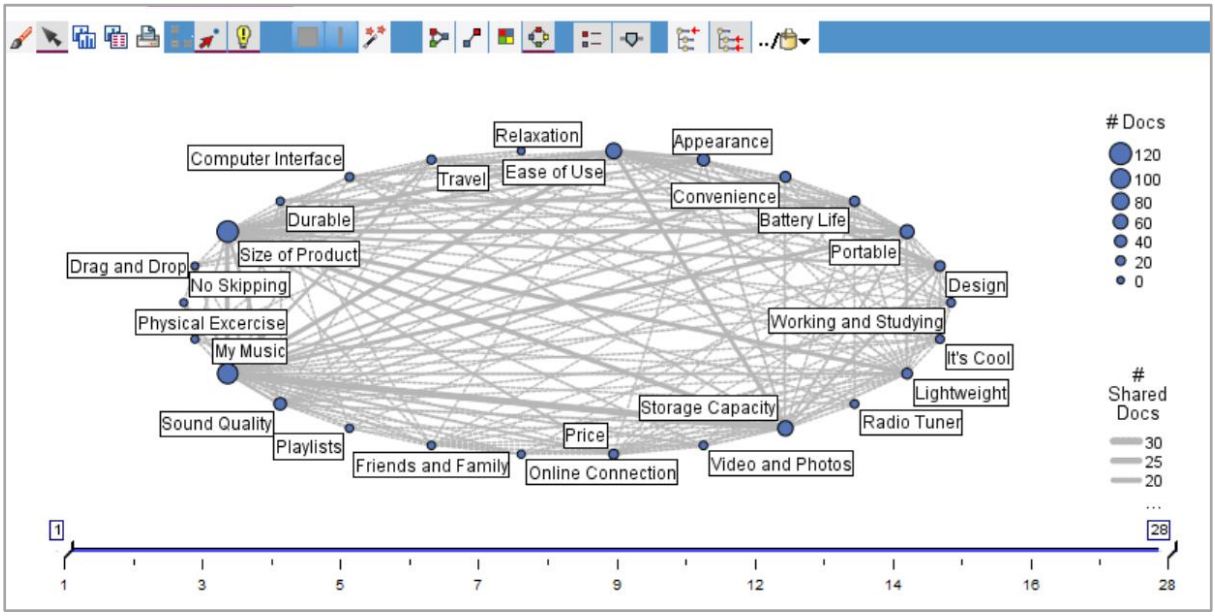


Figure 8.25 Web plot in circular display mode with slider controls

We can see that the plot contains a of detail. The size of the circles within the chart relates to the number of records within that category, whereas the line thickness indicates the number of records where the categories co-occur within the same response. To filter out the weaker connections shift the left-hand slider to position:

5

This changes the display so that each line represents a connection between categories that co-occurs within at least 5 records. You can also move the categories around so these relationships can be seen more clearly.

As Figure 8.26 shows, there is a strong three-way relationship between **Ease of Use**, **Storage Capacity** and **Size of Product**. There is also a clear but weaker relationship between **Battery Life** and **Durable**.

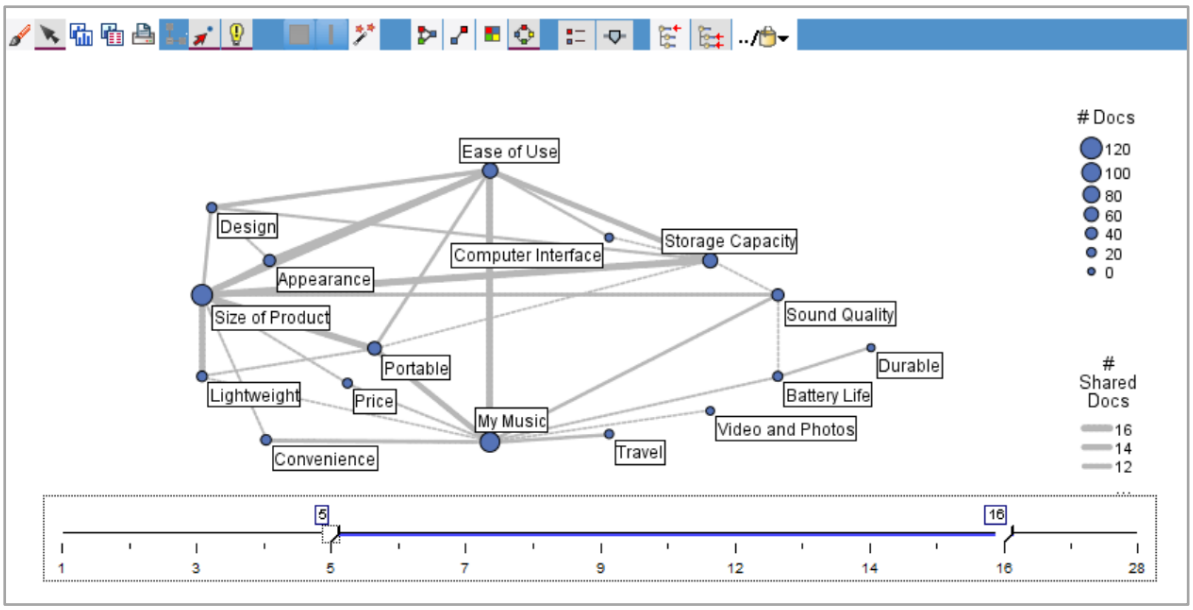


Figure 8.26 Filtered web plot showing category co-occurrence with a frequency of 5

Further details of category co-occurrence can be viewed by clicking the tab marked:

Category Web Table

Here we can see the same information that the unfiltered web plot showed except in tabular form. Looking at the first row in Figure 8.27, within the column marked **Category 1**, the category **My Music** occurs 106 times in the data (as indicated by the associated number in parentheses). Meanwhile, the column marked **Category 2** shows that **Storage Capacity** occurs in 64 records. But the first column labelled **Count** shows that 28 records belonged to the **My Music** category *and* the **Storage Capacity** category simultaneously.

Count	Category 1	Category 2
28	My Music(106)	Storage Capacity(64)
18	My Music(106)	Size of Product(116)
16	Size of Product(116)	Storage Capacity(64)
15	Portable(53)	Size of Product(116)
15	Ease of Use(68)	Size of Product(116)
15	Ease of Use(68)	My Music(106)
14	Lightweight(28)	Size of Product(116)
13	My Music(106)	Portable(53)
13	Ease of Use(68)	Storage Capacity(64)
12	Appearance(40)	Ease of Use(68)
11	Design(25)	Ease of Use(68)
9	Size of Product(116)	Sound Quality(42)
9	Convenience(28)	My Music(106)
8	My Music(106)	Travel(15)
8	My Music(106)	Sound Quality(42)
8	Ease of Use(68)	Portable(53)
8	Design(25)	Size of Product(116)
7	Design(25)	Storage Capacity(64)
7	Computer Interface(12)	Ease of Use(68)
7	Battery Life(22)	Durable(7)
7	Appearance(40)	Size of Product(116)
6	Price(21)	Size of Product(116)
6	My Music(106)	Price(21)

Figure 8.27 Category Web Table showing category co-occurrence in the visualisation pane

8.7 Updating the TAP file

Finally, we can use the newly imported categories to update the existing TAP file. To do so, from the main menu click:

File

Text Analysis Package

Update Package

Within the dialog navigate to the **Data** folder and select:

MP3_Survey.tap

You can see when you click on the file name, that in the column headed **Current Category Set**, the previous category set labelled **Manual Categories** is still present. In the column marked **New Category Set** edit the label so it reads:

Imported Categories

Figure 8.28 shows the dialog at this stage.

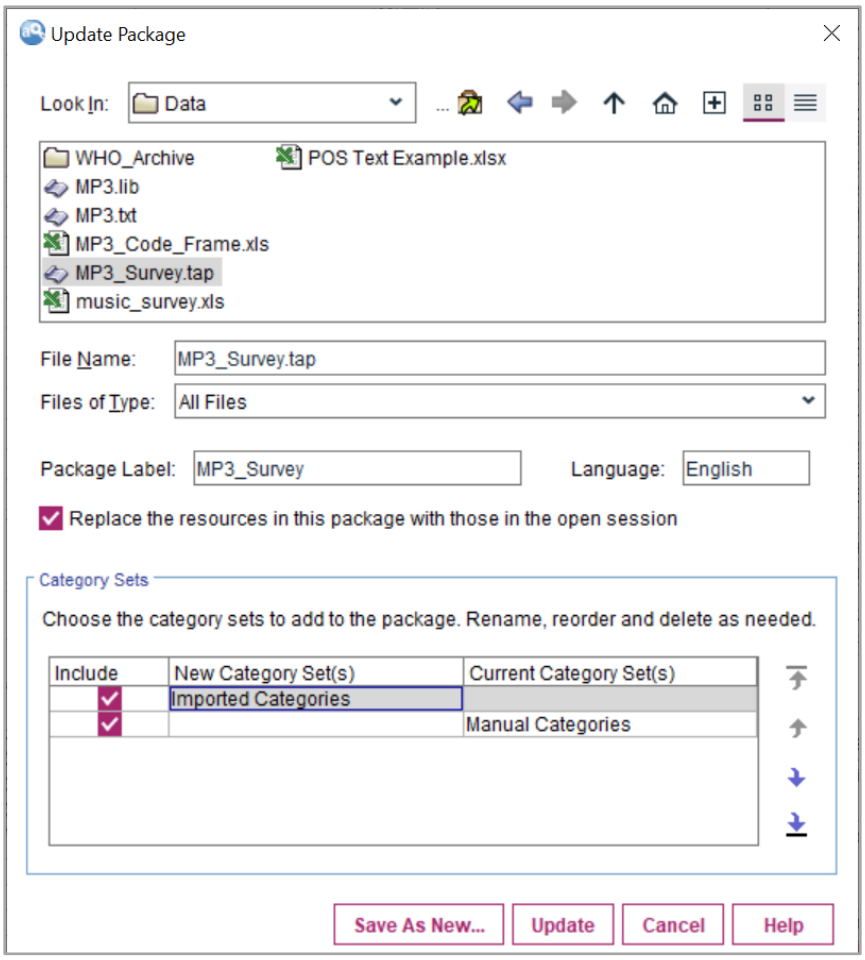


Figure 8.28 Updating the TAP file 'Mp3_Survey.tap'

It's worth noting that this dialog offers a lot of control over the contents of Text Analytics packages. For example:

- To add a new category set from the current session to the TAP file, select the checkbox for the category set to be added.
- If however you want to remove a category set from the TAP file, uncheck the set's corresponding **Include** checkbox. You might do this because you're adding an improved version of the categories.
- One can also replace the linguistic resources inside the TAP with those in the current session by selecting the **Replace the resources in this package with those in the open session** option. This is already selected by default as the current linguistic resources were used to extract the concepts and patterns that defined the current categories. If this option is unselected, the linguistic resources that were already in the package remain unchanged.
- If you only wanted to update the linguistic resources without adding new categories, you could simply ensure this option was selected and choose *only the current category sets* that were already in the TAP for inclusion.
- You can also use this dialog to rename and re-order category sets in the TAP file.
- Clicking the **Save As New** button allows the user to create a new package containing the current session's contents merged with the contents of the selected TAP file. Doing this will then cause the **Save As Text Analysis Package** dialog to appear.

To complete the process so that we have both category sets saved within the TAP file, click:

Update

A message appears asking is you want to replace the existing version of the **MP3_Survey.tap** file. Click:

Yes

The TAP file is now updated.

Now close the session completely by clicking:

File

Close

Exit

Practice Exercise – Chapter 8

Within the folder **Student Exercises** open the following stream:

Chapter_08_Practice.str

1. Right-click on the text mining node and load the resource template **Car Rental** that you created earlier.

After the extraction process has completed, we can experiment with different automatic categorisation settings. For each iteration in this exercise, you should note which categories:

- Are of limited value and should be deleted.
- Appear sensible or useful and worth retaining.
- Could be merged with others to create more meaningful groups.
- Have descriptors which could be deleted or reassigned to other categories.
- Should be renamed.

The aim is to use the automatic categorisation to generate 'easy wins' so that maybe manually created categories could be added later. In which case, you should work towards creating a final set of categories even if they only match with a limited number of records.

In each iteration, delete the categories from the previous run.

2. Go to the **Categories** menu and click **Build Settings**. For simplicity in each of the following iterations, click the **Advanced Settings button** and choose **Flat Categories (single level only)**.

Build from types	Grouping techniques	Maximum categories	Generalize with wildcards
Default	Default (Both)	Default (30)	Default (No)
All	Semantic Network	Default (30)	Default (No)
All	Concept Inclusion	Default (30)	Default (No)
All	Concept Inclusion	45	Default (No)
All	Concept Inclusion	45	Yes

3. Return to the **Build Settings** dialog and choose the **Use frequencies to build categories** mode and follow the same process as previously using the following criteria.

Generate category descriptors at:	Minimum record/doc count
Default (Concepts)	Default (10)
Default (Concepts)	5
Types	10
Types	5

4. Finally, return to the **Build Settings** dialog and using the **Use frequencies to build categories** mode choose the:

- **Generate category descriptors at** the **Concepts level** and
- set the **minimum record/doc count** to **10.**

This time do not delete the categories but return to the **Build Settings** dialog and choose:

- **Generate category descriptors at** the **Types level**
- Also select the **Category Input** setting of **Unused extraction results**

The resultant categories are now *a mix* of those generated by **concepts** *and* **types**.

5. Having tried various categorisation methods choose a final set of categories and add them to the Text Analytics Package file you created earlier.

From the **File** menu, choose **Text Analytics Package** and **Update Package**. Click on the **Car_Rental.tap** file and in the **New Category Set(s)** box, type the label **Automatic Categories** then click **Update.**

6. Finally, delete your final categories and from the **Categories** menu, select **Manage Categories** and **Import Predefined Categories**, browse the **Data** folder and select **Car_Rental_Predefined_Categories.xlsx.**

In the subsequent wizard, choose the following options:

- **Indented format: indentation denotes hierarchy** and click **Next**
- **Extend categories by deriving descriptors**
- **Finish**

When the categories are imported click the **Score** button to see if any cases match the category descriptors. When you have finished, exit the interactive workbench session without updating the node.

Chapter 9 Text Link Analysis

So far in this course, we have made passing reference to text link analysis (TLA). You may recall that the model tab of the text mining node itself contains an option to allow the user to begin the session in the default mode of extracting concepts to build categories or to start by performing text link analysis (see Figure 9.1).

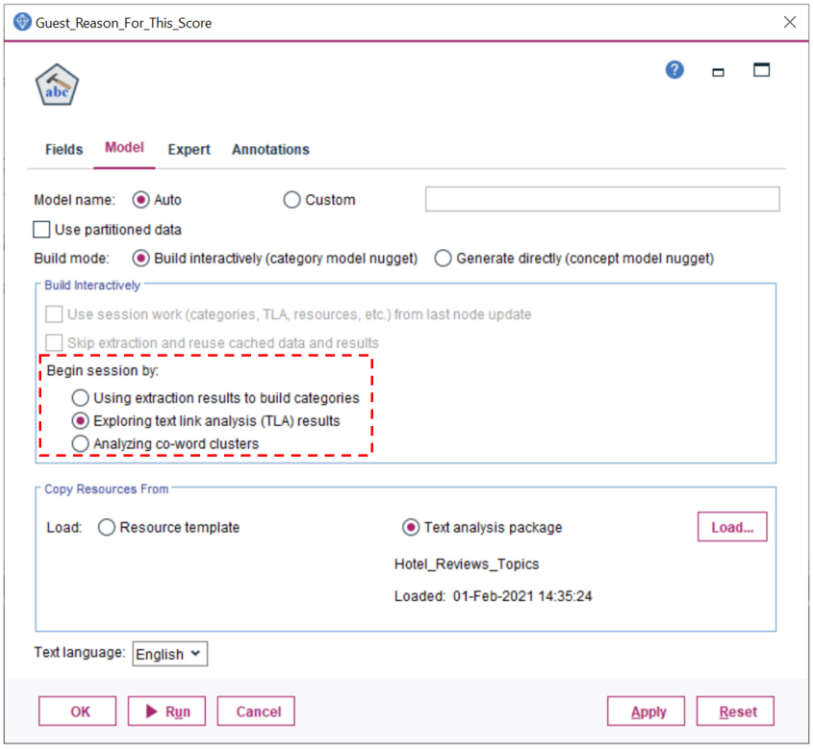


Figure 9.1 Expert tab of the text mining node offering the option to begin the session with text link analysis (TLA)

Text link analysis is a pattern-matching technology that enables users to uncover pattern rules relating to the relationships between types and concepts in the text. You may be wondering how these pattern rules differ from the kinds of category rules that we have already explored. To help illustrate this, the stream file **09_TLA_1.str** contains two stream branches. As Figure 9.2 shows, both branches are reading text from the same Excel data file containing 399 hotel reviews. Customised resources and pre-defined categories have already been saved in a TAP file so the data can be extracted and categorised when either branch is run. The top branch however, begins the session using the default concept extraction method whereas the bottom branch performs text link analysis.

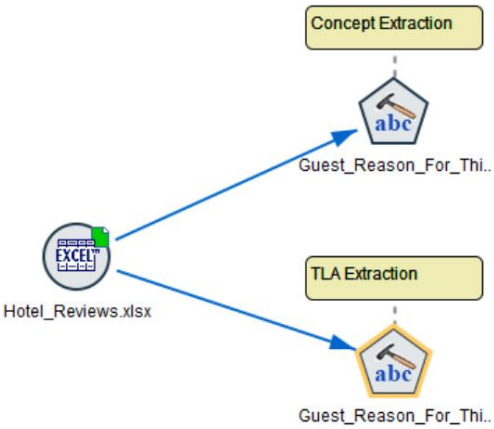
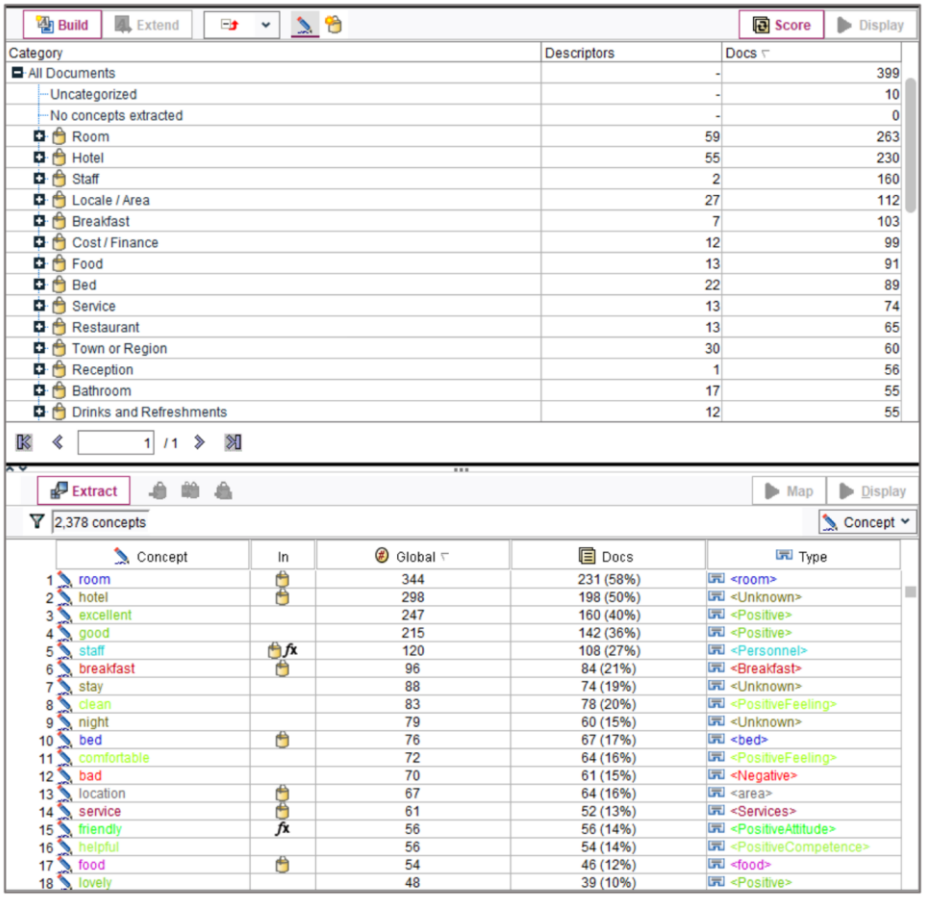


Figure 9.2 Stream file '09_TLA_1.str' containing two text mining nodes performing standard concept extraction and TLA extraction separately

9.1 Relationships in the categories and concepts window

If we run the top branch first, we will see the concepts and categories window appear with the pre-specified categories ready for scoring.



Build | Extend | Score | Display

Category	Descriptors	Docs
All Documents	-	399
Uncategorized	-	10
No concepts extracted	-	0
Room	59	263
Hotel	55	230
Staff	2	160
Locale / Area	27	112
Breakfast	7	103
Cost / Finance	12	99
Food	13	91
Bed	22	89
Service	13	74
Restaurant	13	65
Town or Region	30	60
Reception	1	56
Bathroom	17	55
Drinks and Refreshments	12	55

1 / 1

Extract | Map | Display

2,378 concepts | Concept

	Concept	In	Global	Docs	Type
1	room		344	231 (58%)	<room>
2	hotel		298	198 (50%)	<Unknown>
3	excellent		247	160 (40%)	<Positive>
4	good		215	142 (36%)	<Positive>
5	staff		120	108 (27%)	<Personnel>
6	breakfast		96	84 (21%)	<Breakfast>
7	stay		88	74 (19%)	<Unknown>
8	clean		83	78 (20%)	<PositiveFeeling>
9	night		79	60 (15%)	<Unknown>
10	bed		76	67 (17%)	<bed>
11	comfortable		72	64 (16%)	<PositiveFeeling>
12	bad		70	61 (15%)	<Negative>
13	location		67	64 (16%)	<area>
14	service		61	52 (13%)	<Services>
15	friendly		56	56 (14%)	<PositiveAttitude>
16	helpful		56	54 (14%)	<PositiveCompetence>
17	food		54	46 (12%)	<food>
18	lovely		48	39 (10%)	<Positive>

Figure 9.3 Results from standard extraction mode showing categories and concepts panes for the hotel reviews data

9.1.1 Mapping relationships between concepts

We've already seen that the category web plots allow us to view the relationships between categories that we've created. In fact, we can perform a similar analysis that will allow us to graph the interrelationships between the concepts themselves. Whenever we select a concept within the concepts pane, the button marked **Map** becomes activated. To show what this does, click the commonly occurring concept:

room

Now click:

Map

A message appears telling us that the system is **Building links**. This is an indexing process that occurs the first time the concept map is requested in a given session and it may take several minutes to complete depending on the number of concepts that were extracted. Figure 9.4 shows the resultant concept map for the concept **room**.

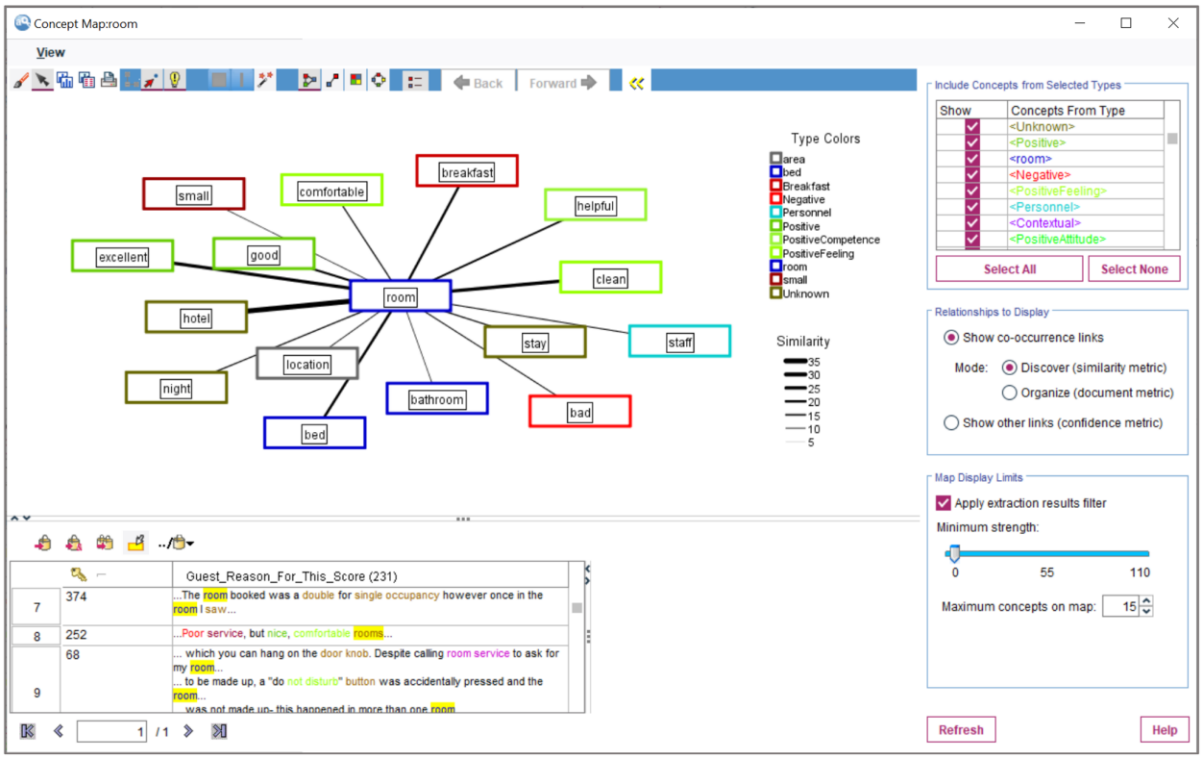


Figure 9.4 Concept map for the concept 'room' showing co-occurrence using the 'Discover (similarity metric)'

The concept **room** is displayed in the centre of the network map surrounded by its most closely-related concepts which are colour coded according to their respective type group. Here the default co-occurrence metric, as shown by the similarity index in the chart legend, is a little more sophisticated than the simple measure used in the category web plot. On the right-hand side of the mapping tool, we can see that the

default co-occurrence link option is labelled **Discover (similarity metric)**. This metric is based on a calculation that encodes how often two concepts appear apart as well as how often they appear together. Higher values indicate that a pair of concepts tend to appear more frequently together than to appear apart. The map indicates that two concepts with the highest similarity scores using the **Discover** algorithm are **hotel** and **clean** with similarity scores of 35 and 25 respectively. Using the **Refresh** button, we can see the text data from the 231 records where the concept **room** occurs. As with the category web plot, using the slider controls here allow us to filter out any weaker relationships.

In this tool, the alternative option for measuring similarity is labelled **Organize** (**document metric**). The **Organize** method measures the strength of the links in the same way as the category web plot i.e., using the raw count of co-occurrences across records. The criticism with this measure is that pairs of the most frequent concepts are likely to co-occur simply because individually, they occur so often to begin with. Figure 9.5 shows that the map is subtly different from the one displayed using previous mode, the link strengths are very different, and the concept **bathroom** has disappeared whereas the concept **friendly** has been added.

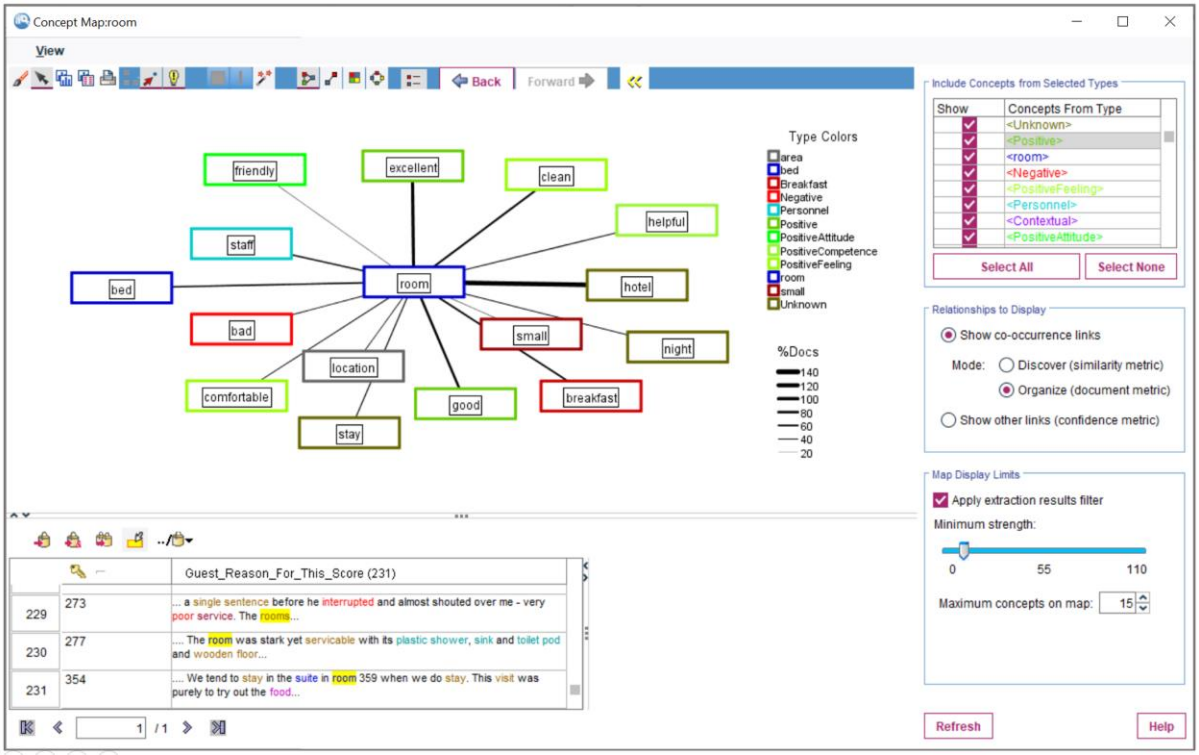


Figure 9.5 Concept map for the concept 'room' showing co-occurrence using the 'Organize (document metric)'

A third way to display relationships with a key concept, is to switch to the mode labelled **Show other links (confidence metric)**. Here the map displays concepts based on their semantic, morphological and syntactic relationships. This approach

calculates the relative link strength based on how many steps removed a concept is from the concept to which it is linked. This can be useful for discovering similar phrases or compound concepts to help with editing and creating resources.

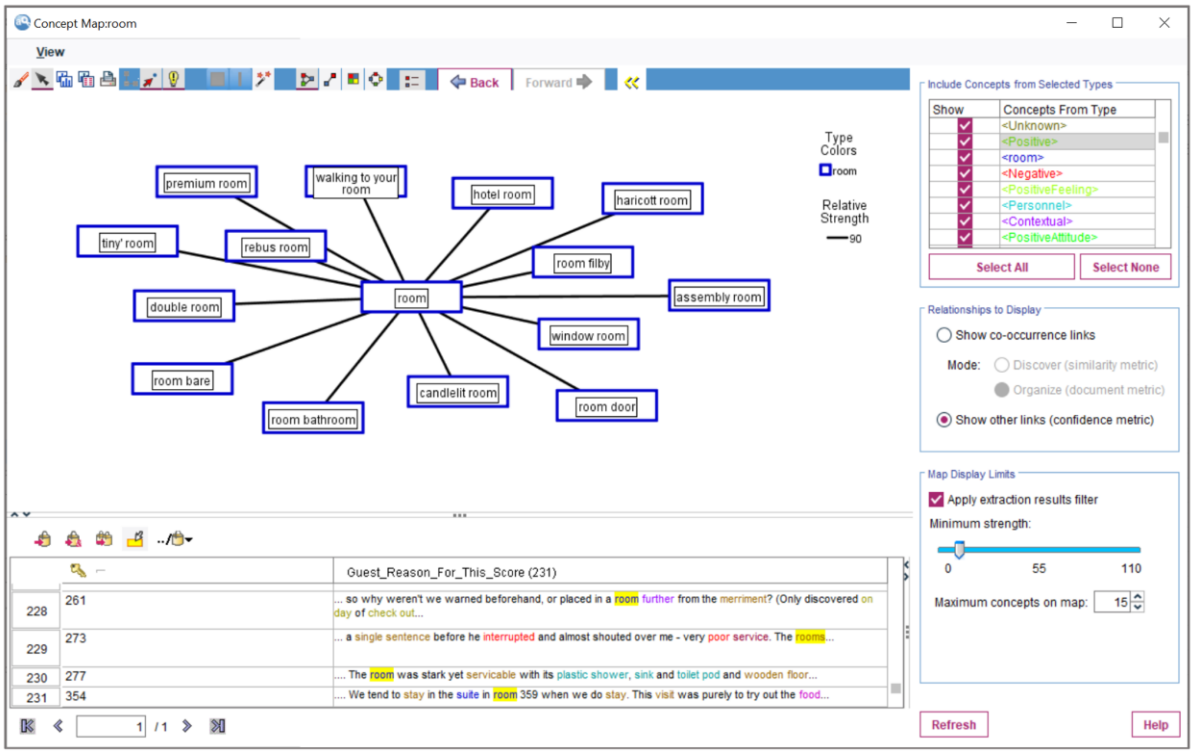


Figure 9.6 Concept map for the concept 'room' show using the 'Show other links (confidence metric)'

Ultimately, the concept map feature is an exploratory tool to help drive insight or the development of better resource files. In a similar way, we can also create categories where concepts co-occur using category rules, but as we shall see, this approach may have certain limitations.

9.1.2 Creating a category rule for co-occurring concepts

Let's assume we want to create a category that captures responses where the hotel guests positively evaluated their hotel rooms. Of course, if we just added the concept **room** to a new category, and then added the concept **excellent** as well, it wouldn't mean that these concepts co-occurred in the same response. It would simply capture responses that contained *either* or both these concepts. We can instead define a rule that populates the category where both positive terms and the concept category occur in the same response. To do so within the category pane:

Right-click and select **Create Empty Category**

Within the resultant pop-up dialog specify the category name:

Good Room

And click:

OK

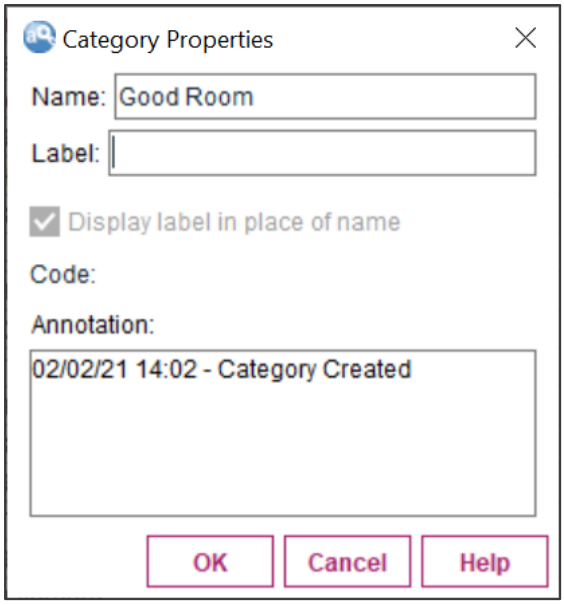


Figure 9.7 Creating an empty category called 'Good Room'

Now the empty category has been created,

Right click on newly created category* Good Room *and select:

Create Category Rule

Within the category rule editor, edit the rule name to read:

Good Room

Now we can define a rule that captures occurrences of the concept **room** as well as the concepts **excellent** or **good**. Note: we might have chosen to use the **positive** type group here instead of the concepts **excellent** or **good**, but for the sake of argument, let's say we wanted to use a narrower range of concepts related to positive sentiments. To include these terms, we can directly drag and drop each concept from the concepts pane into the rule editor, and then use the operator symbols to edit the rule so that it appears as:

room & (excellent | good)

Figure 9.8 shows the rule editor at this stage.

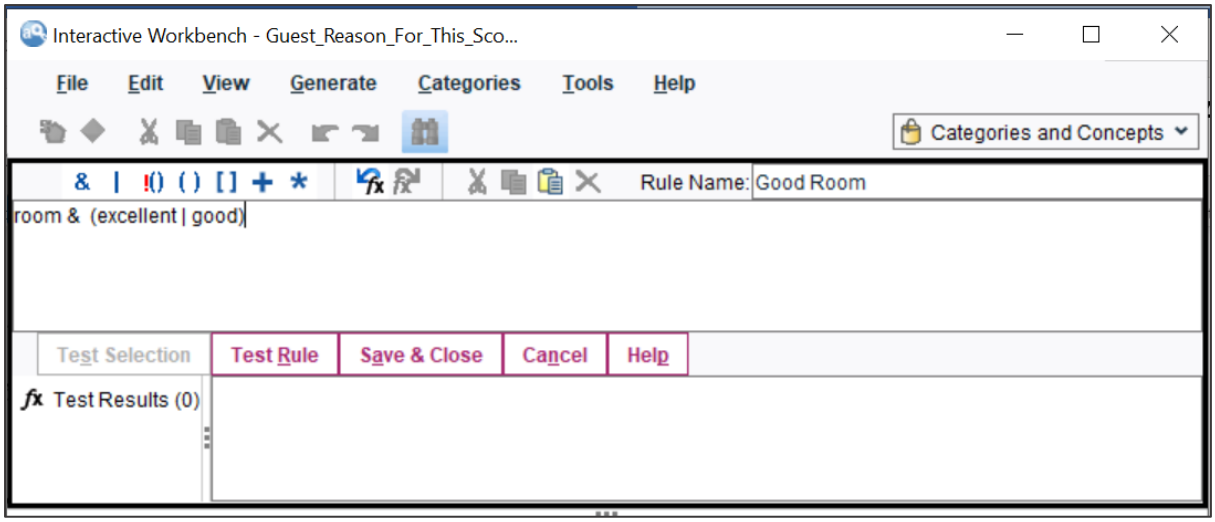


Figure 9.8 Rule capturing co-occurrences of the concept 'room' with the concept 'excellent' or 'good'

To test the rule, within the editor, simply click:

Test Rule

The test indicates that the rule matched with 174 records. Looking at the data pane, the rule initially seems to work well as the first records shows a match with the phrase **great family room**. However, Figure 9.9 shows a screen grab of the second record, with containing two highlighted words that matched with concepts in our rule.

I have stated on a previous review that the staff are excellent and very efficient and happily, this hasn't changed. I also mentioned that the place could do with a bit of a refurb. of soft furnishings,I'd say this is now at a critical stage. The sofa and armchair in our suite were dirty and heavily stained and quite frankly, a disgrace to the industry. Think we'll give this one a miss for a while until they've cleaned up their act.

Figure 9.9 Record containing phrases related to the concepts 'room' as well as 'excellent' or 'good'.

The record illustrates that although the rule correctly identified a response that contained both of our concepts, in reality, the respondent expressed positive sentiments regarding the staff and quite negative sentiment regarding the furniture in the room (or suite). Category rules are therefore better at identifying situations where the category needs to contain a combination of terms or the absence of a key term in order to make sense. Earlier we saw how this approach could be used to discriminate between responses that used the concept **store** to refer to storage versus those that referred to retail. As such, this approach may not be appropriate for sentiment analysis where the subject of the sentiment is of key importance. The

limitation with category rules is that we can't control how *closely* types or concepts co-occur within the same response document. Moreover, *that* degree of control may be particularly crucial when a given record can refer to multiple subjects and multiple sentiments.

To continue our exploration of this topic, within the rule editor click:

Cancel

***Delete the category* Good Room**

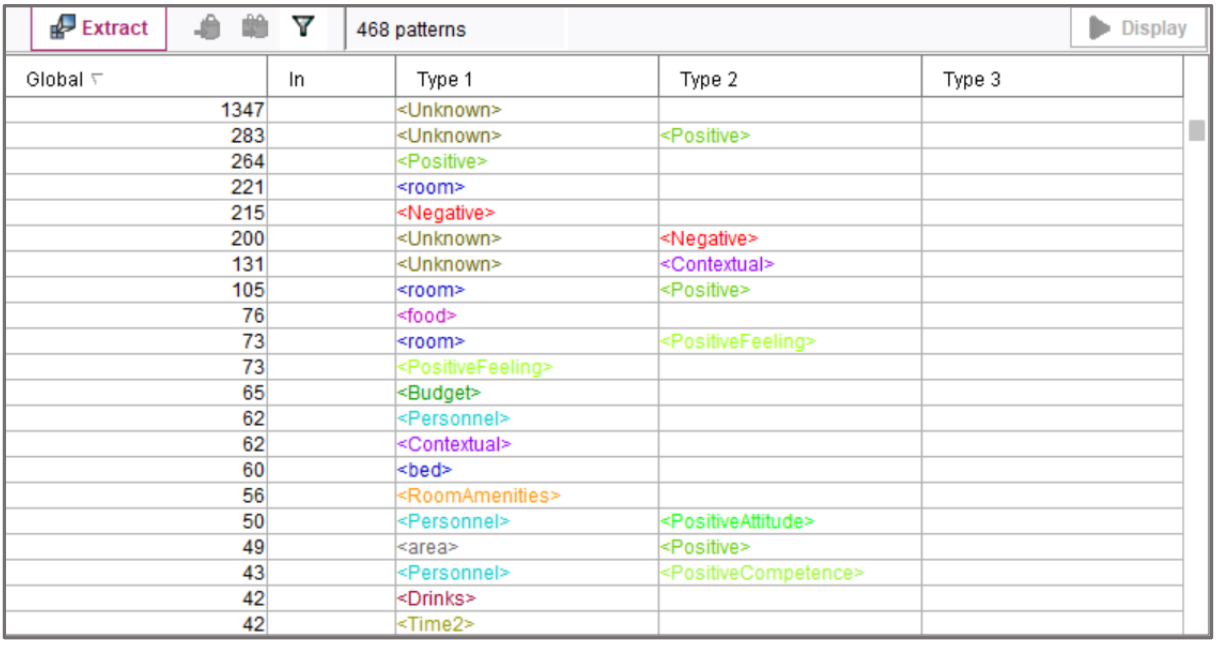
Exit the session without updating the node

9.2 Text Link Analysis

Let's begin this section, by running the second text mining node that labelled:

TLA Extraction

Now when the workbench opens, instead of the categories and concepts window being displayed, we are shown the extraction process occurring in the text link analysis window. We can also see that instead of a categories pane, we are being shown a type patterns pane (see Figure 9.10). These patterns are generated using a default set of rules that actively seek to find relationships between topics and opinions.



Global	In	Type 1	Type 2	Type 3
1347		<Unknown>		
283		<Unknown>	<Positive>	
264		<Positive>		
221		<room>		
215		<Negative>		
200		<Unknown>	<Negative>	
131		<Unknown>	<Contextual>	
105		<room>	<Positive>	
76		<food>		
73		<room>	<PositiveFeeling>	
73		<PositiveFeeling>		
65		<Budget>		
62		<Personnel>		
62		<Contextual>		
60		<bed>		
56		<RoomAmenities>		
50		<Personnel>	<PositiveAttitude>	
49		<area>	<Positive>	
43		<Personnel>	<PositiveCompetence>	
42		<Drinks>		
42		<Time2>		

Figure 9.10 The type patterns pane generated using text link analysis (TLA)

At first the TLA type patterns might appear somewhat vague or uninformative. Firstly, there are a number of single 'slot' patterns where a type is not linked to

another type. Secondly, and rather typically, the most commonly occurring type is the **<unknown>** group. In fact, we can hide the single occurrence type patterns by right-clicking on the pane and from the pop-up menu:

***Uncheck the option* Show One-Slot Patterns**

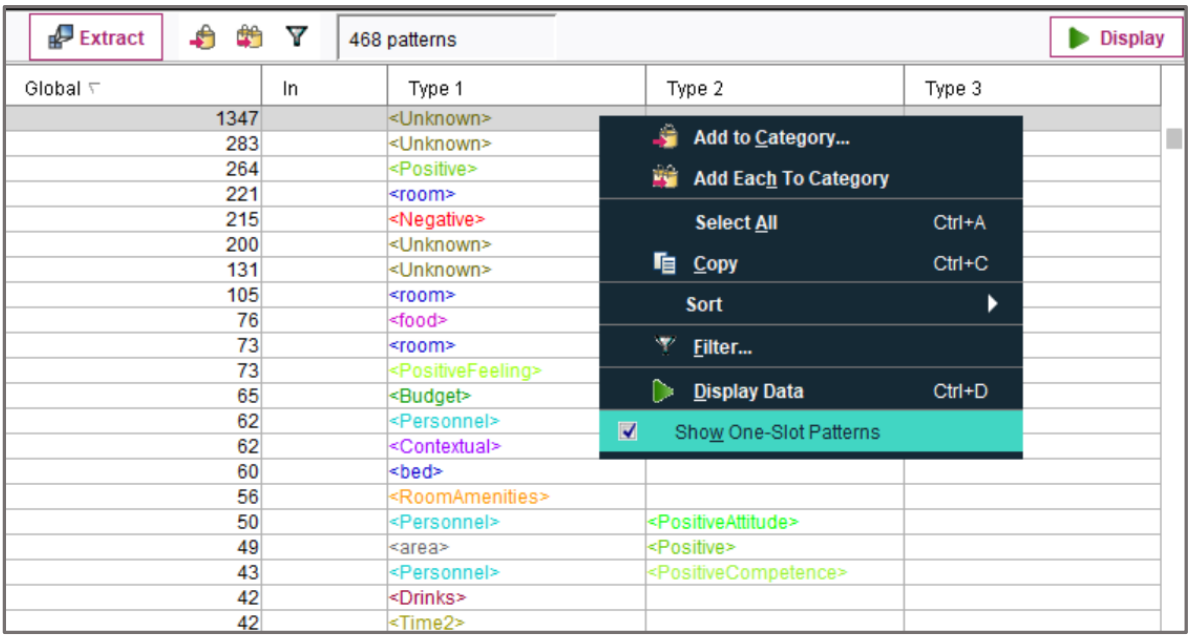


Figure 9.11 Suppressing the display of 'One-Slot Patterns'

If we then click on the most populous type pattern, between **<unknown>** and **<positive>**, we can see that a concept web plot is displayed in the adjacent window. Figure 9.12 shows this with the slider adjusted to a minimum link display of 2.

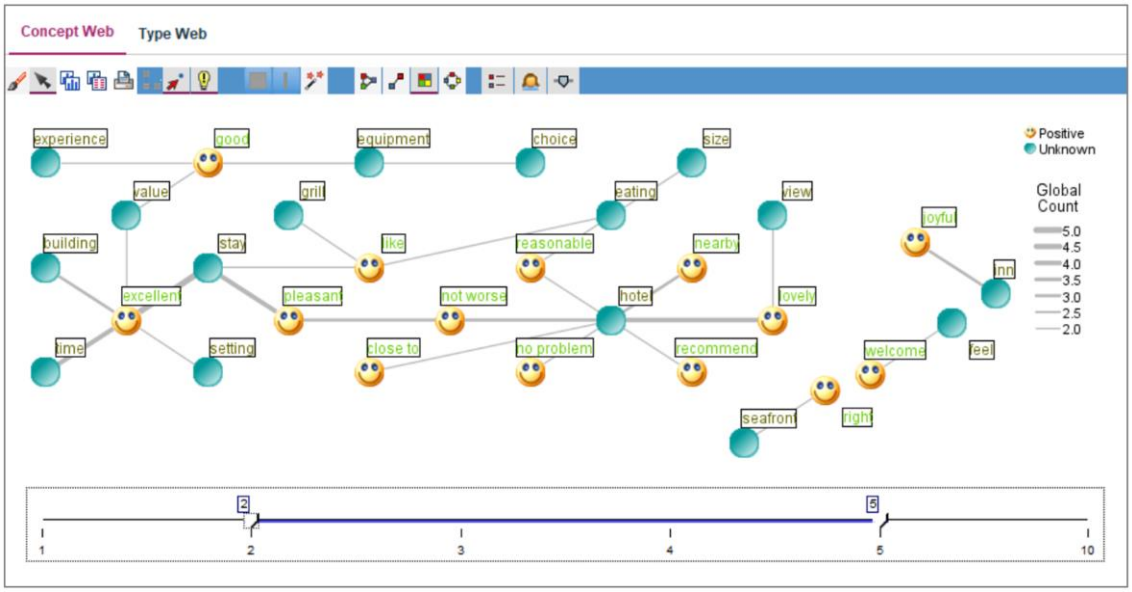
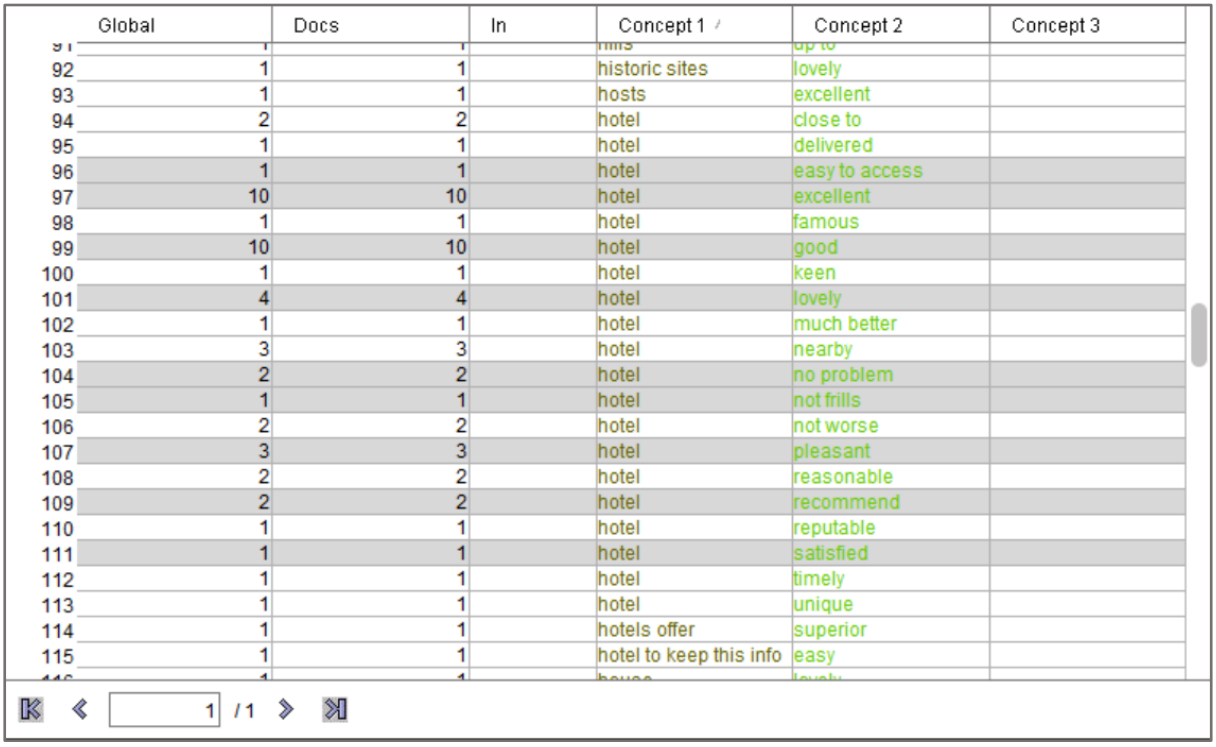


Figure 9.12 Concept web plot showing relationship between 'positive' and 'unknown' concepts

Meanwhile, the bottom left-hand pane shows a list of 226 patterns of cooccurrence between **unknown** and **positive** *concepts*. This could be very useful if we wished to cherry pick the salient interrelationships between these concepts. For example, if we sort the concepts in slot 1 and scroll down, we can see every instance of the concept **hotel** with various associated positive concepts. We can then use either Ctrl-click or Shift-click to highlight the ones of interest as shown in Figure 9.13.



	Global	Docs	In	Concept 1	Concept 2	Concept 3
92	1	1		historic sites	lovely	
93	1	1		hosts	excellent	
94	2	2		hotel	close to	
95	1	1		hotel	delivered	
96	1	1		hotel	easy to access	
97	10	10		hotel	excellent	
98	1	1		hotel	famous	
99	10	10		hotel	good	
100	1	1		hotel	keen	
101	4	4		hotel	lovely	
102	1	1		hotel	much better	
103	3	3		hotel	nearby	
104	2	2		hotel	no problem	
105	1	1		hotel	not frills	
106	2	2		hotel	not worse	
107	3	3		hotel	pleasant	
108	2	2		hotel	reasonable	
109	2	2		hotel	recommend	
110	1	1		hotel	reputable	
111	1	1		hotel	satisfied	
112	1	1		hotel	timely	
113	1	1		hotel	unique	
114	1	1		hotels offer	superior	
115	1	1		hotel to keep this info	easy	

1 / 1

Figure 9.13 Concept patterns sorted alphabetically by a lead term with patterns of manually interest selected

If we click the display button to reveal which records match with these patterns, we can see that 34 responses are returned, the vast majority of which do indeed exhibit positive sentiment with regard to the concept **hotel**. As Figure 9.14 shows, this is a more accurate way to uncover sentiment and topic relationships than using manually defined category rules following the default method of extracting concepts. Moreover, it is possible to simply right-click on highlighted selection and request that a new category is created based on the selected patterns.

		Guest_Reason_For_This_Score (34)
1	205	...All in all a good hotel...
2	226	...I can recommend this hotel in the beautiful city of Bath to anybody who wants a break,...
3	154	...Really impressed with this hotel after only paying around £50 for a double "compact" room...
4	366	...for the hotel - professional, pleasant and keen to help...
5	3	.... other than that a lovely hotel, great location...
6	318	...Great hotel...
7	59	...Hotel is pleasant and staff helpful and cheerful...
8	78	.... Would come again and recommend the hotel...absolutely....
9	284	...Two nights spent in this hotel with absolutely nothing to complain about and too much to praise...
10	215	...First of all, I would like to say the hotel is easy to find, close to...
11	320	...Had a wonderful weekend here, The hotel is really lovely, with lots of quirky areas...
12	165	...Stayed in this Absolutely great hotel, really helpful staff, they made our stay very pleasant...
13	258	...should start by saying the hotel itself was fine...
14	344	...Nice clean hotel in a great location, staff are friendly & the numerous "good...
15	1	.... We moved from first room owing to unpleasant smell , hotel did not quibble and moved us but still slight unpleasant smell...
16	4	...A really lovely hotel...
17	239	...lovely unusual hotel, great setting and good service...
18	198	...We were very happy with our stay at the Rebus Bristol Centre a bustling no-frills hotel...
19	188	...but in summary this is a comfortably pleasant hotel if all else fails...
	45	...Nice hotel, well situated, in the centre of bristol near the shopping centre and

Figure 9.14 Responses returning a match between the concept 'hotel' and the selected positive concepts

When the type group has already been clearly identified, this approach might even be more straightforward. Figure 9.15 shows that selecting the type **<room>** and the types **<Positive>** and **<PositiveFeeling>** returns 114 records where people expressed positive approval of their rooms. The same approach finds several other useful sentiment-topic combinations as summarised in the table in Figure 9.16.

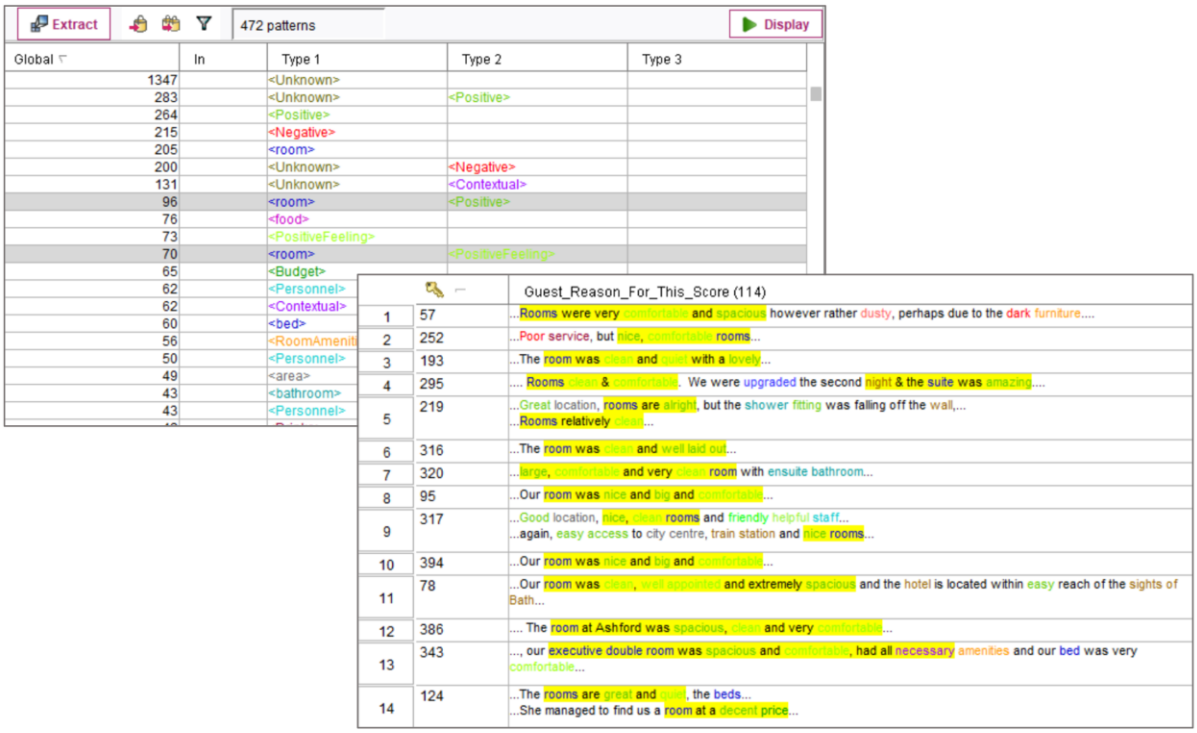


Figure 9.15 Matches between the type '<room>' and the selected types '<Positive>' and '<PositiveFeeling>'

Slot 1 type group	Slot 2 type group(s)	Records matched
<room>	<Positive> <PositiveFeeling>	114
<personnel>	<Positive> <PositiveFeeling> <PositiveAttitude> <PositiveCompetence>	89
<area>	<Positive>	49
<room>	<Negative> <NegativeFeeling> <NegativeFunctioning> <small> <Poor Hygiene>	35
<bed>	<Positive> <PositiveFeeling>	34
<reception>	<Positive> <PositiveFeeling> <PositiveAttitude> <PositiveCompetence> <PositiveRecommendation>	27

Figure 9.16 match rate between types using TLA extraction

9.2.1 Automatic Categorisation with TLA

It's possible to automatically create categories using the default TLA patterns. To show this, we will return to the categories and concepts window. For the sake of clarity:

Select and delete all the existing category groups

Now from the main menu click:

Categories

Build Settings

The settings dialog is once again displayed. This time, the default mode is to build categories from type patterns. We can also see that the option **Structure categories by pattern type** is activated. This is a constraint, in that any generated patterns are limited to those of a specific type group. Moreover, the default type is **<Unknown>**. If we wish to generate categories based on descriptors that match across at least two slots in the type patterns, we can deselect the first type pattern which consists just of the **<Unknown>** type group on its own. Figure 9.17 shows the dialog at this stage.

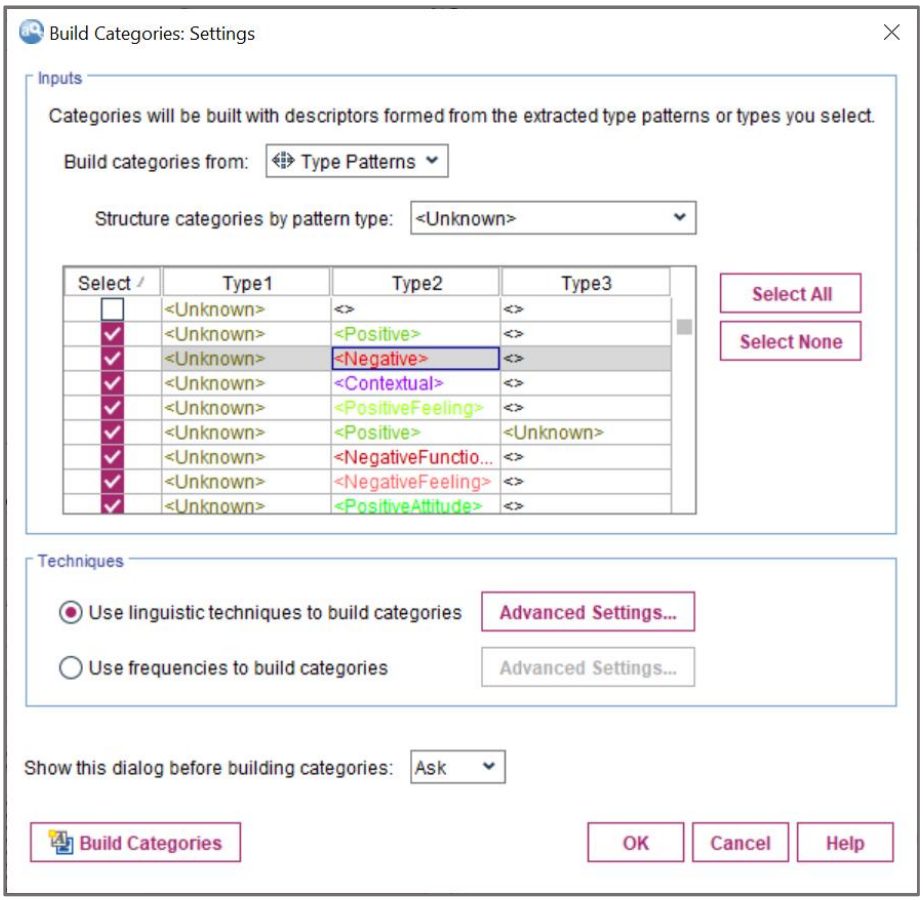


Figure 9.17 The Build Categories: Settings dialog configured to create categories based on type patterns

To generate the categories, click:

Build Categories

Unfortunately, the categories are built in a hierarchical fashion meaning that the first category, **hotel**, contains a number of descriptors that effectively refer to miscellaneous aspects of the concept hotel, followed by six sub-categories focussed on the concept **hotel** and various opinions types such as **<Negative>** and **<Positive>** (see Figure 9.18).

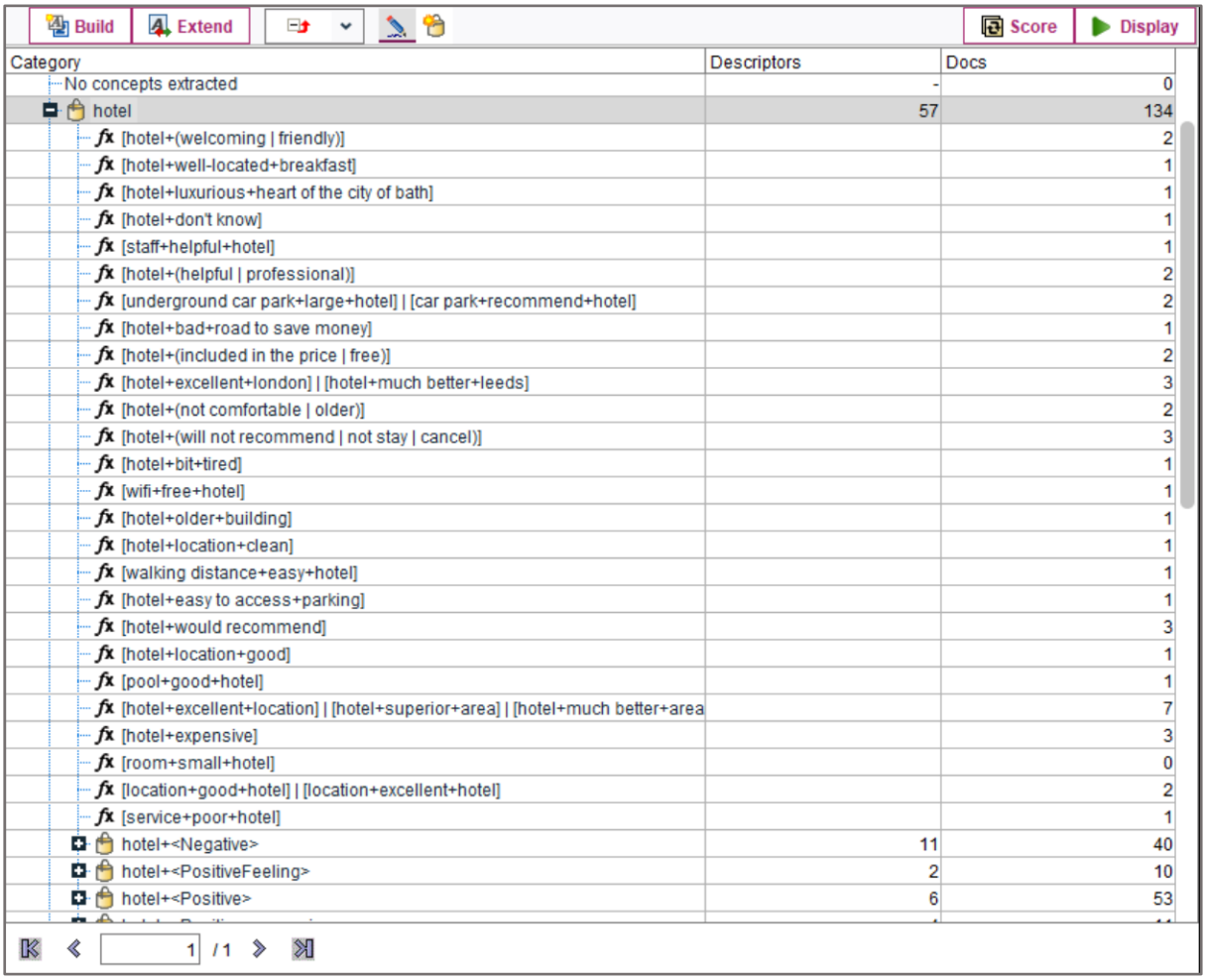


Figure 9.18 Categories generated from type patterns and structured by the <Unknown> type

Requesting, flat categories would not help here as the process would simply merge all the categories into one. One approach would be to highlight the sub-categories of interest and move them to the top of the hierarchy as shown in Figure 9.19 but this might prove to be quite onerous, and we should bear in mind that these categories are all built with respect to only one type group **(<Unknown>)** so the entire process might have to be repeated for other types as well.

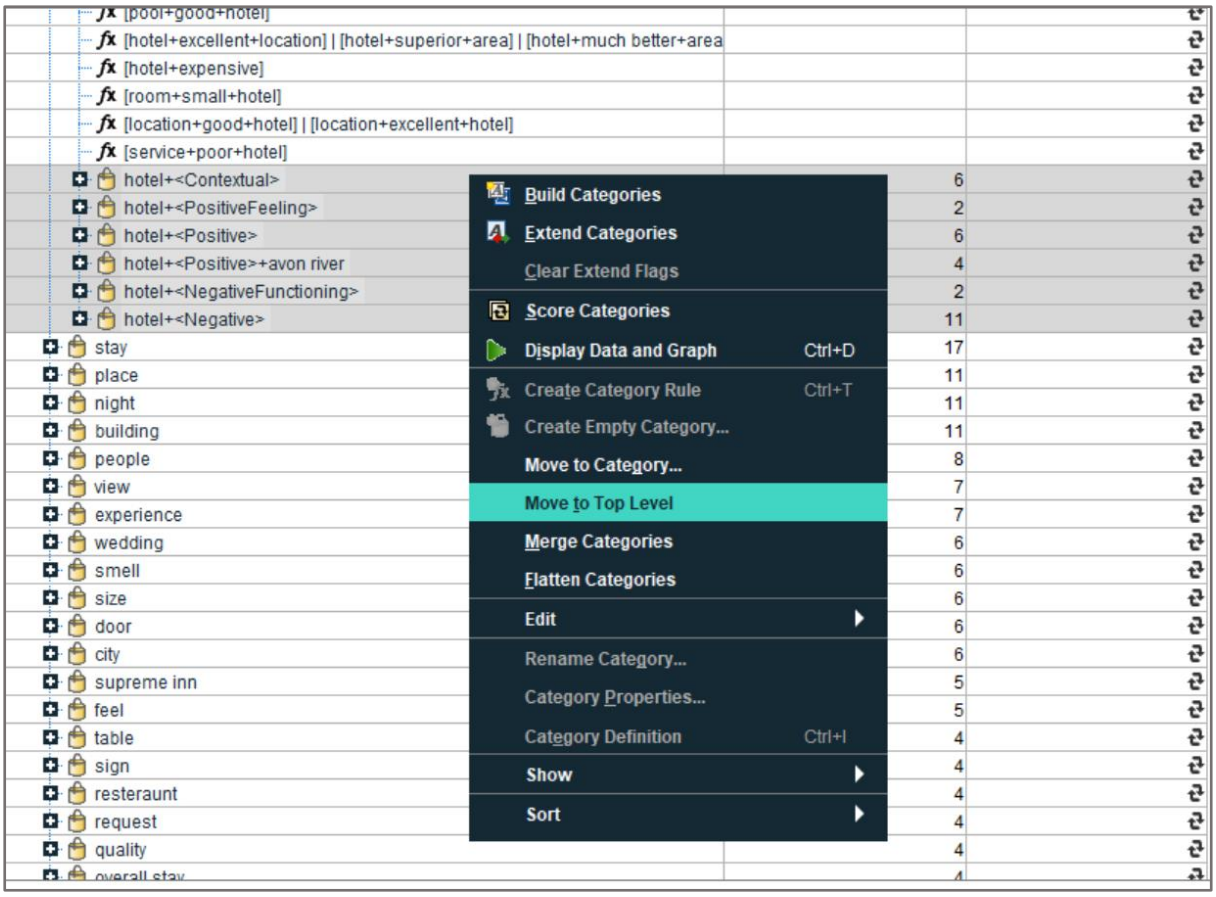


Figure 9.19 Moving sub-categories to the top level of the hierarchy

9.3 Creating custom text link analysis rules

So far, we have discovered that if we run text link analysis extraction using resources that include libraries like the Opinions library, we can choose individual text link patterns and use them to create categories that not only refer to particular topics but indicate *how* respondents refer to those subjects (i.e., positively or negatively). We also saw that we could run the automatic categorisation procedure and choose to retain or merge text link analysis categories that were of interest to us (although this might be quite time-consuming).

In this section, we are going to look at where the default text link analysis pattern rules can be accessed and edited within the system. Before we do so:

Select and delete all current categories in the category pane

Now navigate to the:

Resource Editor

Within the resource editor, click the tab marked:

Text Link Rules

The Text Link Rules editor is displayed and shown in Figure 9.20.

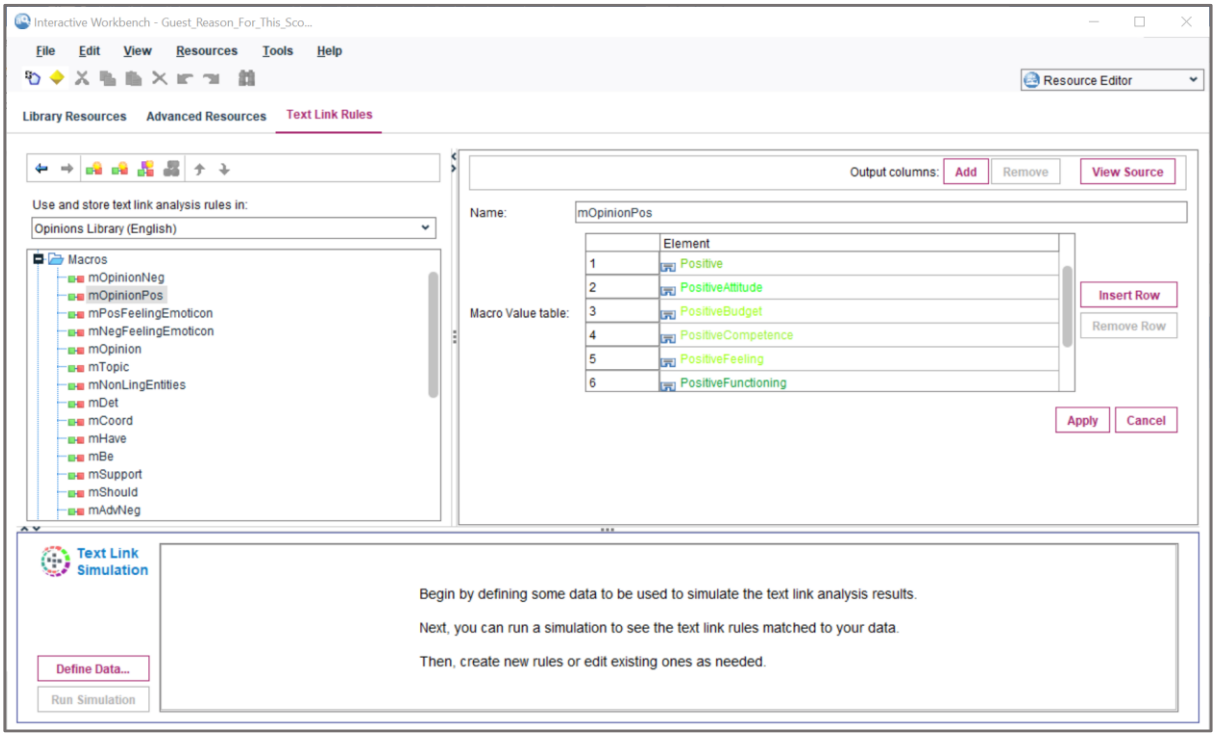


Figure 9.20 Text Link Rules editor with positive opinions macro displayed

The text link rules editor provides a number of useful functions, such as:

- Viewing and editing existing text link rules
- Disabling or enabling specific rules
- Defining which terms, concepts and types are included in rules
- Creating and editing new custom rules
- Controlling how rules are executed within the system
- Creating simulated rules with sample text

Most text link rules are driven by two key elements.

1. **Macros**: just as types contain concepts and terms, macros are containers that can hold text strings (specific words or phrases), existing types or even other macros. Macros make it easier to create rules that refer to multiple types or terms. Macro names should be prefixed with the letter **m**.
2. **Rules**: we can create text link rules that allow us to capture occurrences of text in a specific order containing particular words, types or macros. For example, we might be interested in responses that include type groups such as **staff**, **manager** or **reception** followed by a positive concept, such as, **the waiters were attentive**, or **reception were very professional**.

Figure 9.21 shows the list of default macros associated with the Opinions library on the left-hand side of the editor. Note that it also shows the contents of the macro **mBe**, this contains a list of words (not types) that are associated with the verb **to be**.

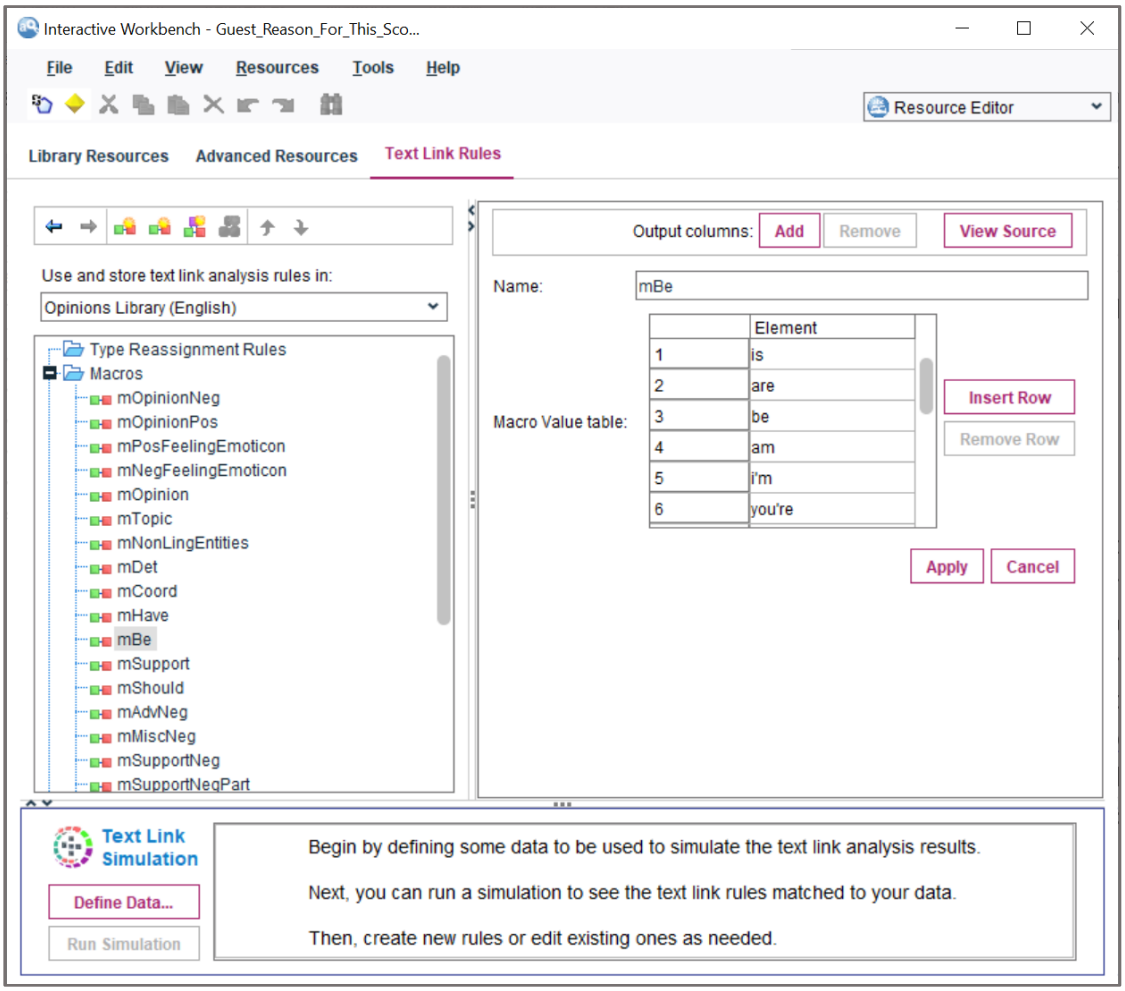


Figure 9.21 the list of default macros associated with the Opinions library – the macro 'mBe' is selected

If we scroll down this section of the editor, we can see the extensive list of default text link analysis rules. These rules can be edited, copied or disabled and they govern every aspect of the text link analysis process with Modeler Text Analytics. The rules all belong to a single, pre-defined rule set from the Opinions library called **1_OPINIONS**. Figure 9.22 shows a specific rule for dealing with negative phrases containing positive terms. The example used to test the rule is the phrase **it was not a good hotel**. Here the rule syntax is designed to deal with the fact that although the text mentions a topic followed by the positive concept **good**, it is negated by the negative word **not**. The resulting text link analysis pattern returns a single slot outcome for the type group **negative**.

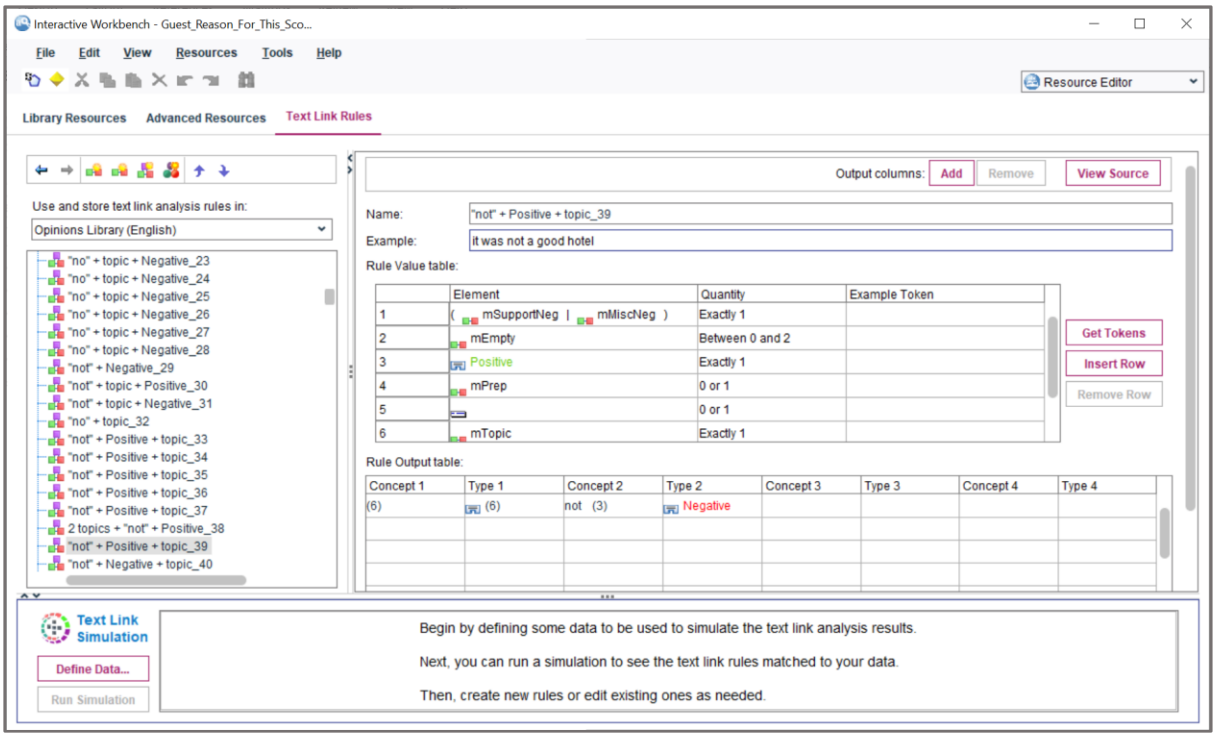


Figure 9.22 Text link analysis rule designed to deal with negated positive terms

This part of the resource editor also contains a **Text Link Simulation** tool that helps users understand how sample phrases can be used to design rules. As an example, in the bottom left-hand corner click the button marked:

Define Data

A **Simulation Data** dialog is generated. Within the dialog type the phrase:

the staff were great

Figure 9.23 shows the dialog at this stage. To run the simulation and see how the text link pattern rules evaluate the phrase, click:

Run Simulation

It is recommended that users only enter single phrases using this tool as it can take a prohibitively long time to evaluate multiple phrases.

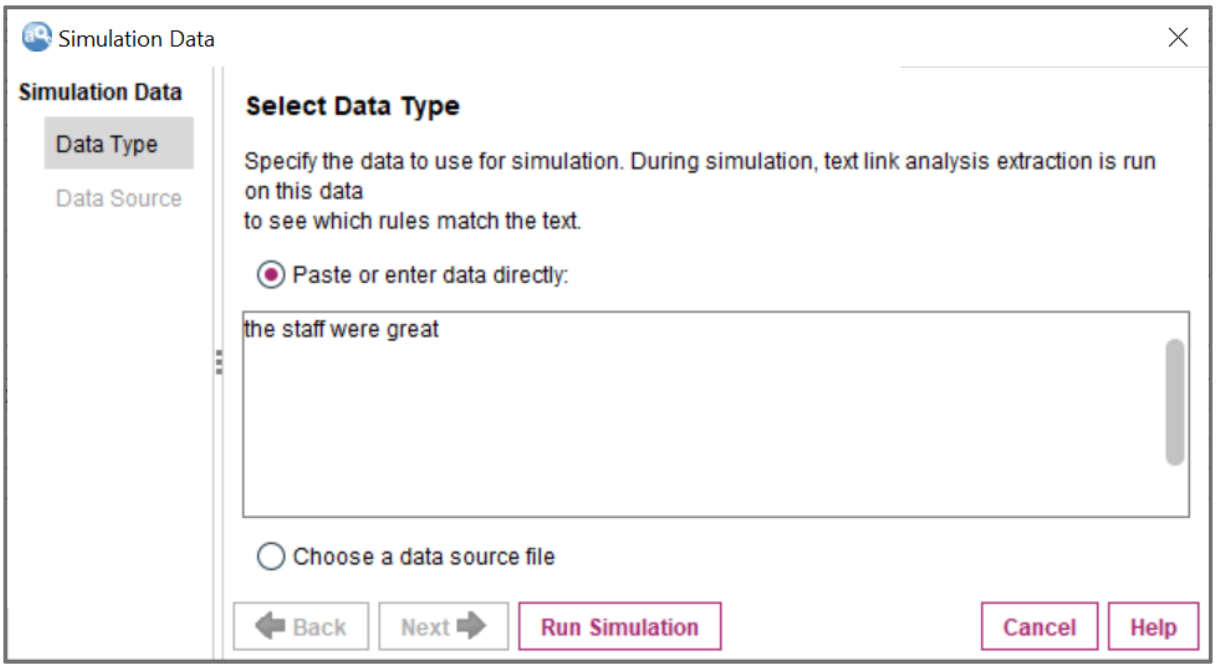


Figure 9.23 Simulation data dialog used to suggest text link rule syntax

The output generated by this simulation is shown in Figure 9.24. The input text is broken down by the matching macros and type groups as follows:

- **the –** this word does not belong to a type group but it is recognised by the macro **mDet** which identifies determiners like **a**, **an** and **the**
- **staff –** this word belongs to the type group **Personnel** but it isn't part of a macro
- **were –** this word does not belong to a type group but does belong to the macro **mBe** which contains words like **is**, **are** and **were**
- **great –** this word is matched with the type **Positive** and belongs to the macro **mOpinionsPos**

Figure 9.24 Matching types and macros from the phrase 'the staff were great'

Figure 9.25 shows what this rule would actually generate as part of the text link analysis output. The rule output box shows that the phrase matches a rule in the **1_OPINIONS** rule set called **topic + opinion_190**. This rule generates a two-slot pattern where the types are **Personnel** and **Positive** and underlying concepts are **staff** and **excellent**.

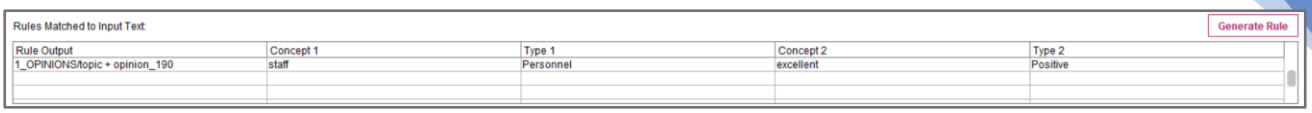


Figure 9.25 TLA outputs from the sample rule matching the phrase 'the staff were great'

In bottom right-hand corner we can click the button marked **Generate Rule** to request that the system automatically creates a new rule specifically for this sample of text. Figure 9.26 shows the generated rule syntax for our sample text.

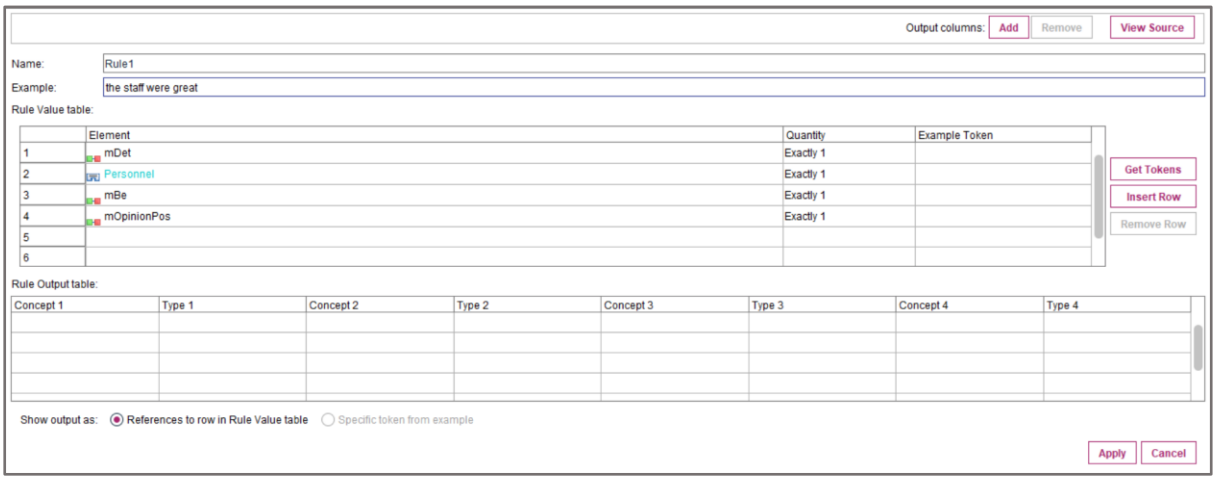


Figure 9.26 TLA automatically generated rule from sample text

We can actually view the source code beneath the rule by clicking the button marked:

View Source

Figure 9.27 shows the underlying source code.

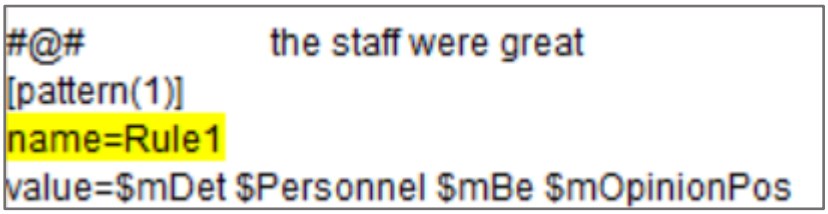


Figure 9.27 Source code for example rule

You may notice that the source code for this rule simply lists the syntax elements of the rule with each element prefixed by a **$** sign and separated by a space. The rule name is highlighted in yellow to indicate which rule the code refers to. To return to the previous window, click:

Exit Source

To specify the rule outputs, we can simply click and drag them from the rule value table into their respective slots. Alternatively, right-click in the cells of the outputs table and select the elements. Figure 9.28 shows the populated outputs table. Note that within this table the first output slots relating to **Concept 1** and **Type 1** are simply numbered **(2)** indicating that they will be populated with words from the type **personnel**, which is the *second* syntax element in the example text. In the same fashion, the slots in **Concept2** and **Type2** will be occupied by the fourth element in the phrase, which are words contained in the positive macro group.

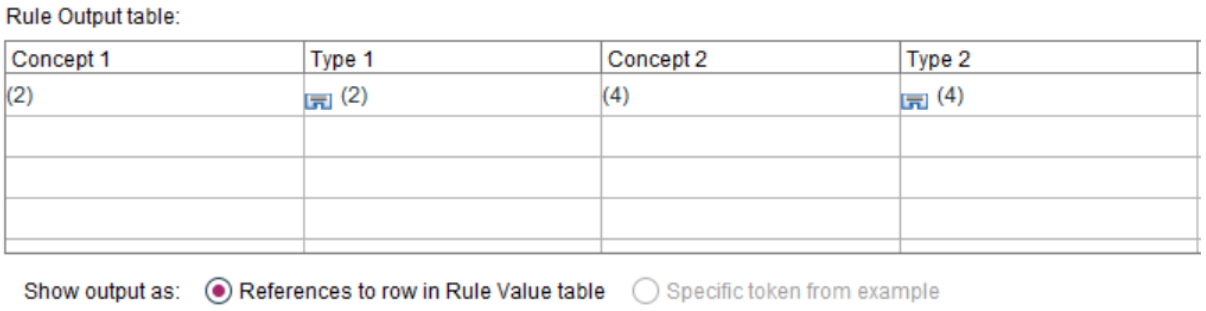


Figure 9.28 TLA rule outputs from our sample phrase

To test the rule, on the right-hand side click the button:

Get Tokens

The **tokens** in this procedure refer to the actual words in the sample text which are then compared to the rule syntax and outputs to check if the rule works correctly. After a while, the rule is successfully evaluated, and the output table is updated as shown in Figure 9.29. You should note that this is a sensitive procedure and prone to throwing error messages indicating that the sample text does not match any known rule. If this occurs, make sure the rule output table is correctly specified and that the source code has not been corrupted.

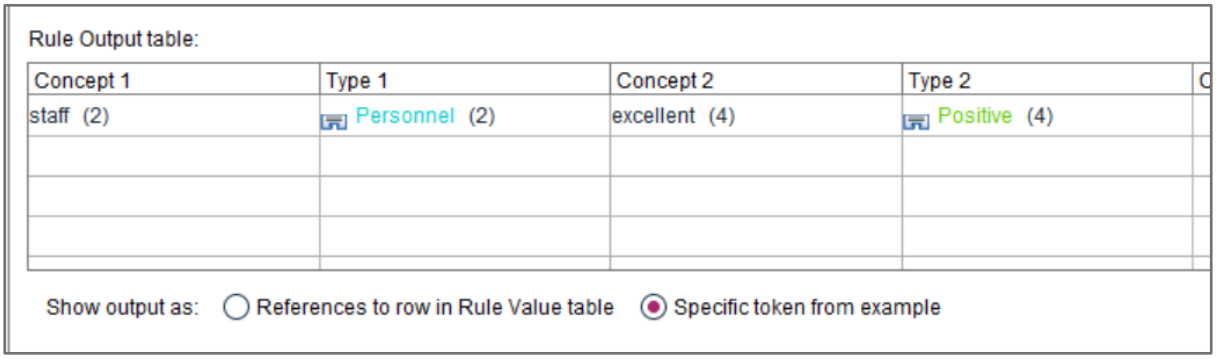


Figure 9.29 TLA rule outputs after the 'Get Tokens' procedure has been executed

This method of creating rules tends to be slightly restrictive in that, in its current form, similar phrases will have to match the syntax exactly in order to be recognised by the rule. As an example, if we edit the phrase in the example box so that it reads:

the staff were really great

And again click:

Get Tokens

We encounter an error message as Figure 9.30 shows. Because the text is slightly different, the current rule syntax does not match the specified rule.

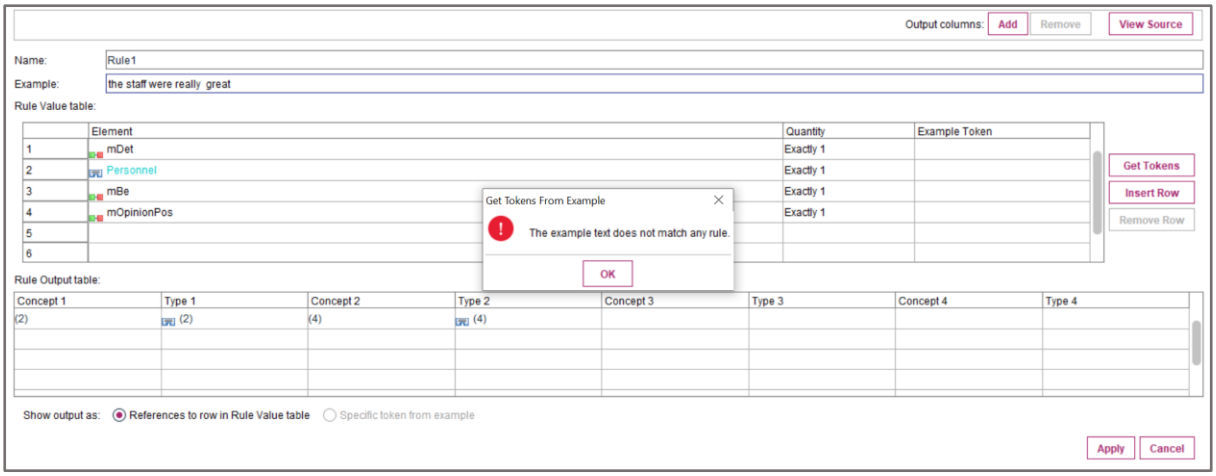


Figure 9.30 Error message following the 'Get Tokens' request due to the example phrase being altered

We can edit the rule syntax to take account of this extra element in the sample text by right-clicking on the row header of the last item in the **Element** column and from the drop-down menu choosing:

Insert Row Above

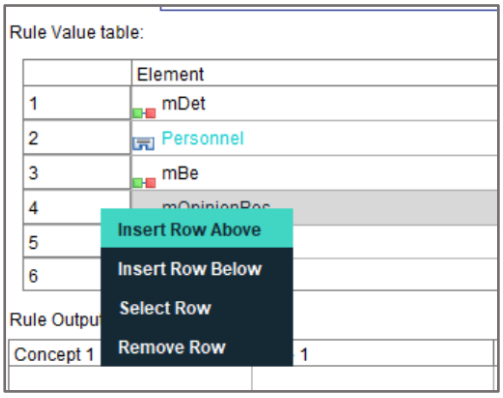


Figure 9.31 Inserting an extra row in the element column of the TLA rule editor

Once the extra row has been inserted, we can right-click on the blank cell and from the drop-down menu choose:

<ANY TOKEN>

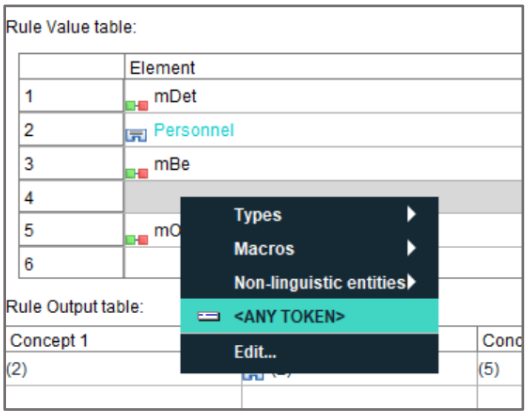


Figure 9.31 Inserting an <ANY TOKEN> in the penultimate row of the element column of the TLA rule editor

The **<ANY TOKEN>** acts as a wildcard token for terms in an expression that we wish to ignore so the rule is evaluated correctly. It's worth noting however, that the word **really** is an adverb, so we might have used the macro **mAdverb** here instead. The advantage of using the **<ANY TOKEN>** element is that can be used to match a broader set of text phrases.

You may also note that the output table is now updated so that **Concept2** and **Type2** now refer to the element **(5)**. Once again, to check the rule, we can click:

Get Tokens

Figure 9.32 shows the updated rule editor.

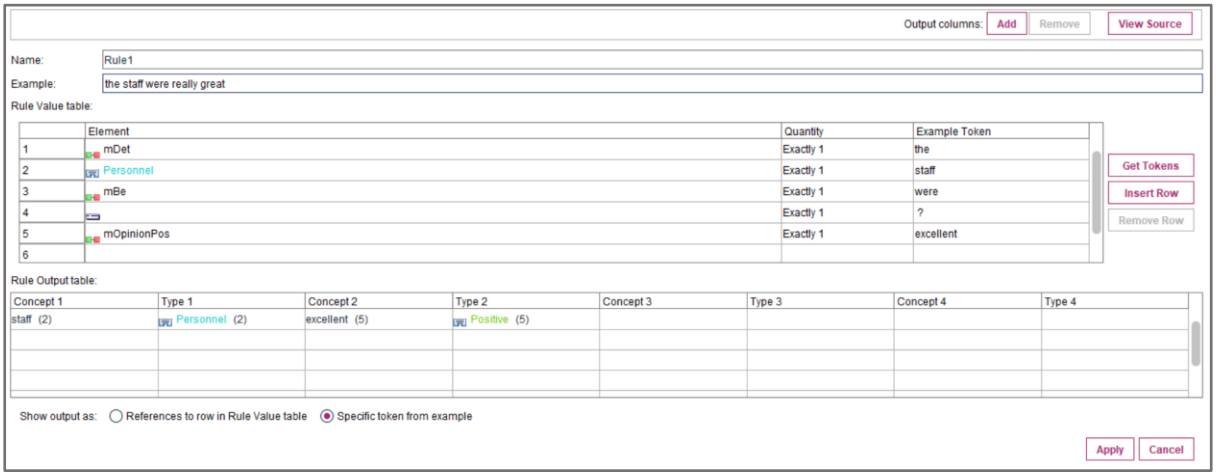


Figure 9.32 The updated TLA rule editor

We can see from the quantity column that the rule is expecting an occurrence of **exactly 1** for the **<ANY TOKEN>** element in the phrase. We could change this value so that it read **Between 0 and 2** so that the following phrases were all acceptable:

the staff were great

the staff were really great

the staff were really really great

Figure 9.33 illustrates this.

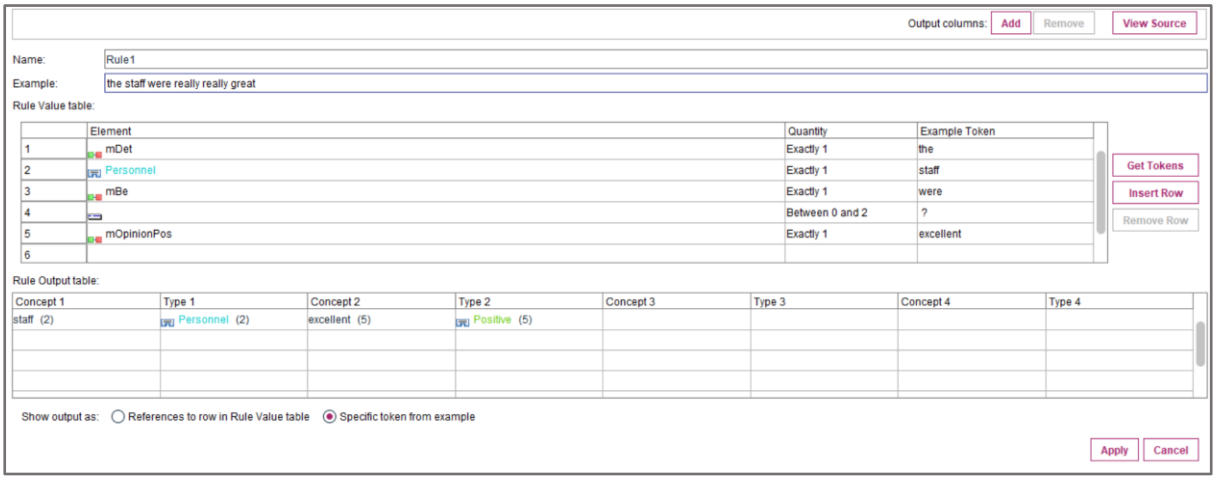


Figure 9.33 Using the 'Between 0 and 2' quantity value with the <ANY TOKEN> element to match a wider range of phrases

In this section, we've seen how using the simulation tool is a handy way to figure out how to structure a TLA rule. This is made easier given the fact the system already contains large collection of macros for incorporating into new rules.

9.4 Creating custom rules

Fortunately, it's not necessary to edit all the existing rules in order to generate a wide range of TLA patterns that show interactions between adjectives and nouns. This after all is at the heart of sentiment-based categorisation, where the analyst attempts to classify text not just in terms of what subjects or topics are mentioned, but also *how* the respondents are talking about them. It's possible therefore for the user to create two large macros with one containing a list of positive and negative types and the other a list of noun topics. A generic rule can then be defined that will find multiple permutations of interactions between the elements of each macro group

To illustrate how one might design their own rules we will begin by disabling the entire set of rules for the Opinions library. To do so:

Right click on the title of the list of rules: **Rule Set 1_Opinions**

From the drop-down menu, click:

Disable

Figure 9.34 shows this process.

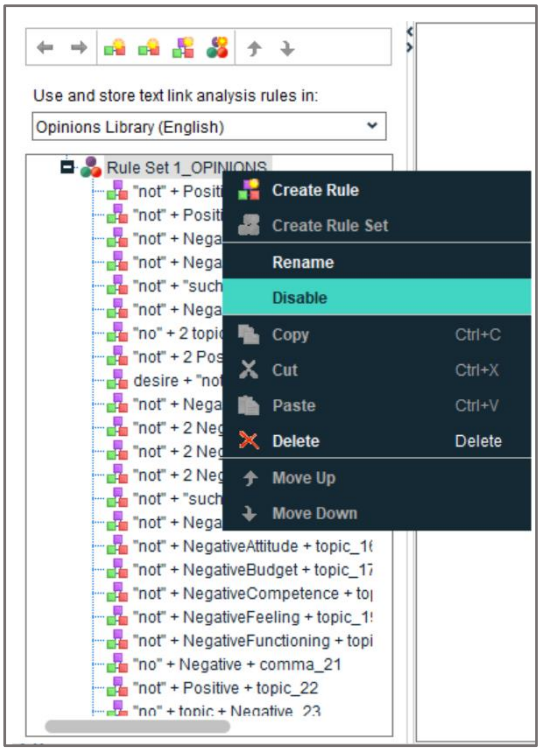


Figure 9.34 Disabling the existing rule set

Now that we have disabled the rules, we will create a new macro group. Within the macro list, right-click, and from the pop-up menu select:

Create Macro

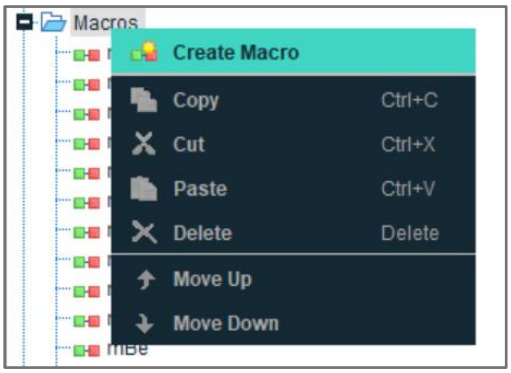


Figure 9.35 Creating a new macro

Call the new macro group:

mSubjects

We can now add a small sample of types to the macro to illustrate how custom rules can be defined. Right-click in the associated macro value table, and from the pop-up menu, add a list of types that refer to noun subjects such as the following examples:

Room

Hotel

Staff

Area

Breakfast

Food

Bed

Figure 9.36 shows this list added to the macro group **mSubjects**.

Figure 9.36 Types added to the macro group mSubjects

Having created an example macro group, we must define a rule. Scroll down to the now disabled rules and right-click to create a new rule set. Call the new rule set:

Hotel_Rules

Figure 9.37 Creating a new rule set

Having created a new rule set, we can now define a rule. Once again, right-click on the rule set **Hotel_Rules** and from the pop-up menu select:

Create Rule

Within the rule editor define a rule with the following properties:

Rule Value table	
Rule Name	Subject_Sentiment
Example text	Breakfast was excellent
Row 1 Element	mSubjects
Row 1 Quantity	Exactly 1
Row 2 Element	<ANY_TOKEN>
Row 2 Quantity	Between 0 and 2
Row 3 Element	mOpinions
Row 3 Quantity	Exactly 1

Rule Output table	
Concept 1	(1)
Type 1	(1)
Concept 2	(3)
Type 2	(3)

Figure 9.38 shows the completed rule.

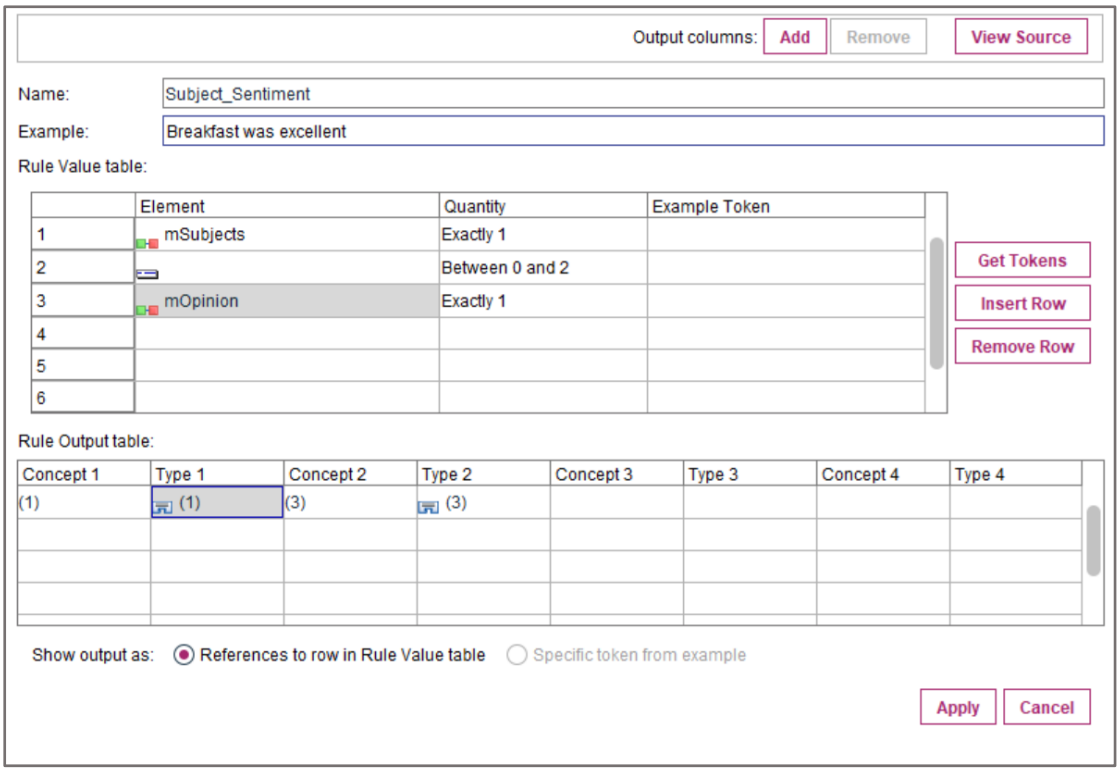


Figure 9.38 Example custom rule

To test the rule against the example text, click:

Get Tokens

Note: Don't be too alarmed if you receive a message stating that the text does not match a rule. This does not mean that the rule won't work. Try altering the example text by experimenting with different words and see if that works.

We can see that in Figure 3.39, our text has been evaluated with the tokens extracted and correctly typed.

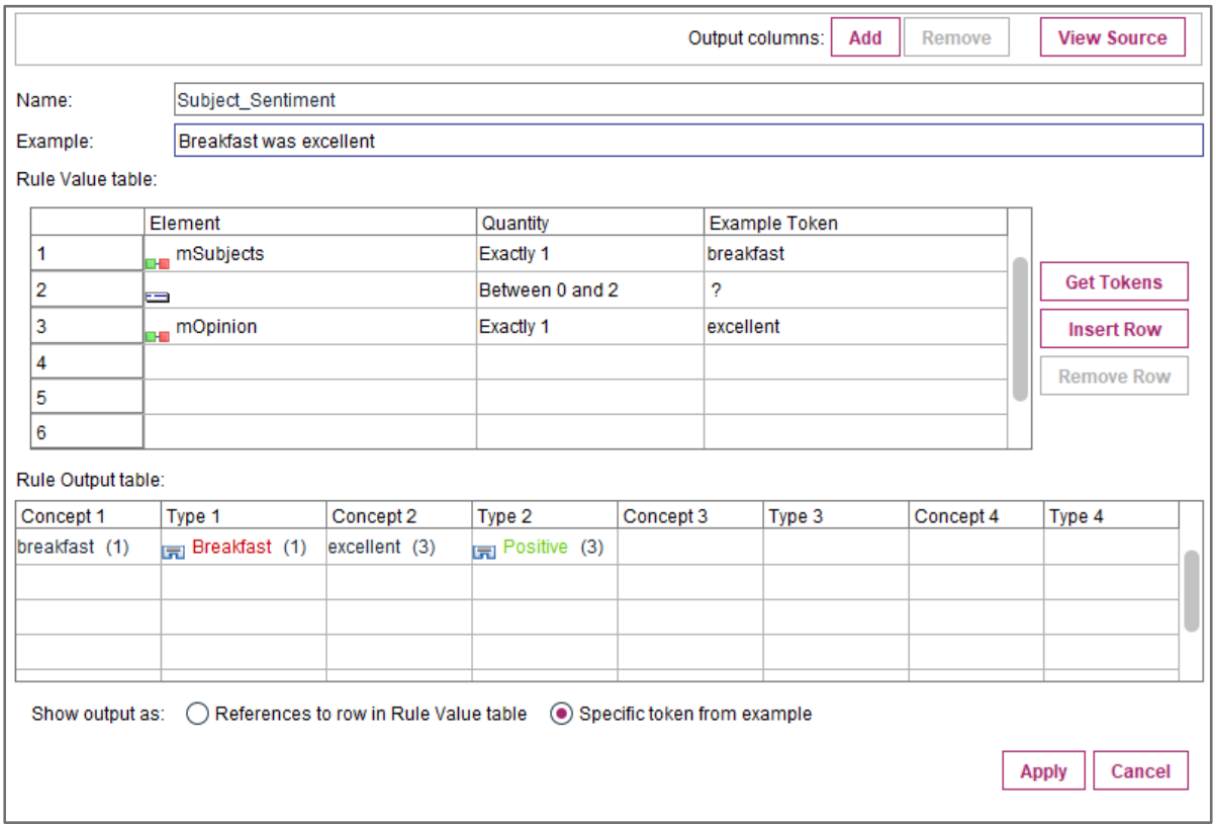
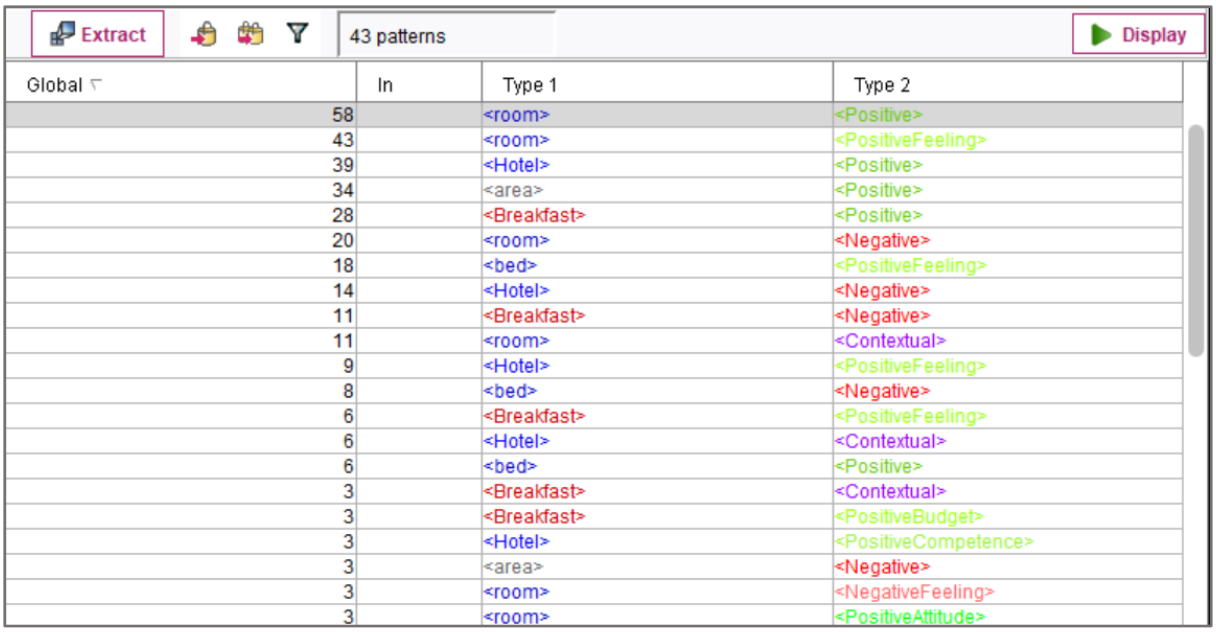


Figure 9.39 Example custom rule – with tokens extracted and typed

Now to test out the rule, return to the Text Link Analysis window and click:

Extract

After the extraction process has finished, we should have a number of rules that detect positive and negative sentiment related to the short list of subjects that we provided. Remember that we need only right-click on any of these to add them to create new categories or add them to existing ones. Figure 3.40 shows an image of the extracted TLA patterns generated by this single rule.



Global	In	Type 1	Type 2
58		<room>	<Positive>
43		<room>	<PositiveFeeling>
39		<Hotel>	<Positive>
34		<area>	<Positive>
28		<Breakfast>	<Positive>
20		<room>	<Negative>
18		<bed>	<PositiveFeeling>
14		<Hotel>	<Negative>
11		<Breakfast>	<Negative>
11		<room>	<Contextual>
9		<Hotel>	<PositiveFeeling>
8		<bed>	<Negative>
6		<Breakfast>	<PositiveFeeling>
6		<Hotel>	<Contextual>
6		<bed>	<Positive>
3		<Breakfast>	<Contextual>
3		<Breakfast>	<PositiveBudget>
3		<Hotel>	<PositiveCompetence>
3		<area>	<Negative>
3		<room>	<NegativeFeeling>
3		<room>	<PositiveAttitude>

Figure 9.40 Extracted patterns from the Subject_Sentiment rule

You can see from the extracted patterns that all the phrases follow the same syntactic direction of subject followed by sentiment. If we return to the rule editor, we can create a copy of our rule and alter it so that we can also capture responses where the sentiment *occurs before* the subject. Figure 9.41 illustrates this new reversed rule, appropriately entitled **Sentiment_Subject**. Note that in the *output table*, the rule has been edited so the subject still occurs before the sentiment. This will mean that when we *extract* the rules again, in most cases we will simply add more records to the existing rules rather than force the creation of new ones.

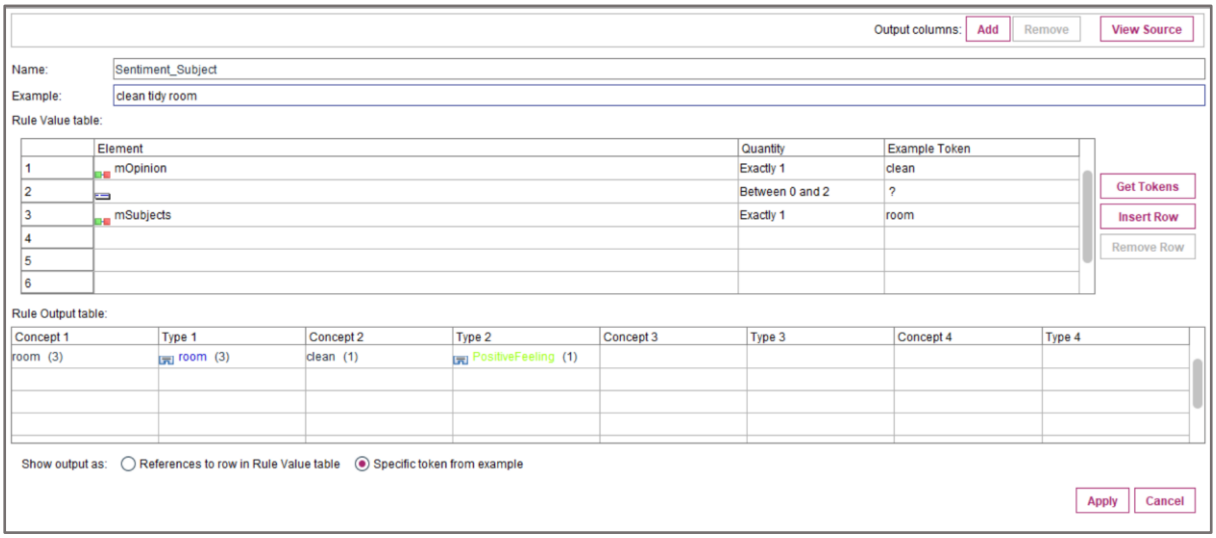
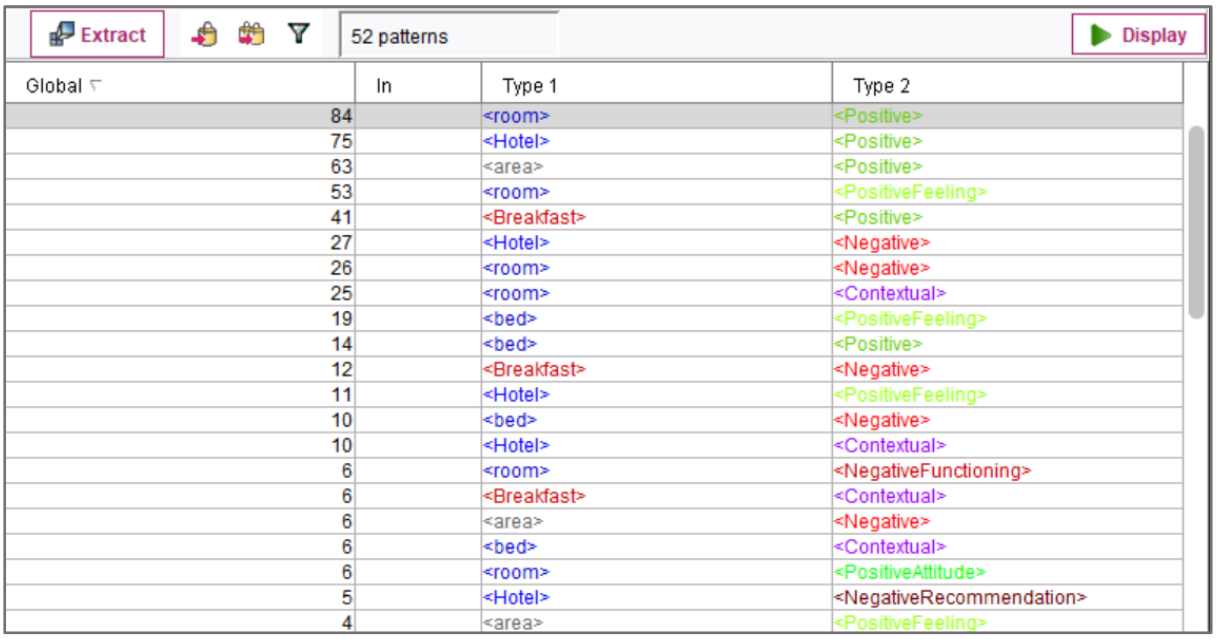


Figure 9.41 Reversed version of the custom rule – with tokens extracted and typed

Figure 9.42 shows the effect of adding this rule to the extraction process. Previously, 58 records were matched with the **<room>** and **<Positive>** rule, but now we've matched 84 records.



Global	In	Type 1	Type 2
84		<room>	<Positive>
75		<Hotel>	<Positive>
63		<area>	<Positive>
53		<room>	<PositiveFeeling>
41		<Breakfast>	<Positive>
27		<Hotel>	<Negative>
26		<room>	<Negative>
25		<room>	<Contextual>
19		<bed>	<PositiveFeeling>
14		<bed>	<Positive>
12		<Breakfast>	<Negative>
11		<Hotel>	<PositiveFeeling>
10		<bed>	<Negative>
10		<Hotel>	<Contextual>
6		<room>	<NegativeFunctioning>
6		<Breakfast>	<Contextual>
6		<area>	<Negative>
6		<bed>	<Contextual>
6		<room>	<PositiveAttitude>
5		<Hotel>	<NegativeRecommendation>
4		<area>	<PositiveFeeling>

Figure 9.42 Extracted patterns from the Subject_Sentiment rule and the Sentiment_Subject rule

We can see immediately, that just by using two rules and a limited number of subjects, we can generate a wide range of sentiment-based patterns that can in turn be used to create categories.

9.5 Applying TLA rules to categories

As a final example in this chapter, we can open a stream with a pre-defined TAP file containing two custom TLA rules that refer to a wider range of subjects. Open the stream:

09_TLA_2.str

Run the text analytics node in the stream and view the extracted TLA pattern results as shown in Figure 9.43. In this stream, the macro group containing the noun subjects is much more extensive, so a larger number of patterns are extracted and more responses are subsequently categorised. As the extraction results show, a number of categories have already been defined and are populated with an extensive list of descriptors provided by the extracted TLA patterns.

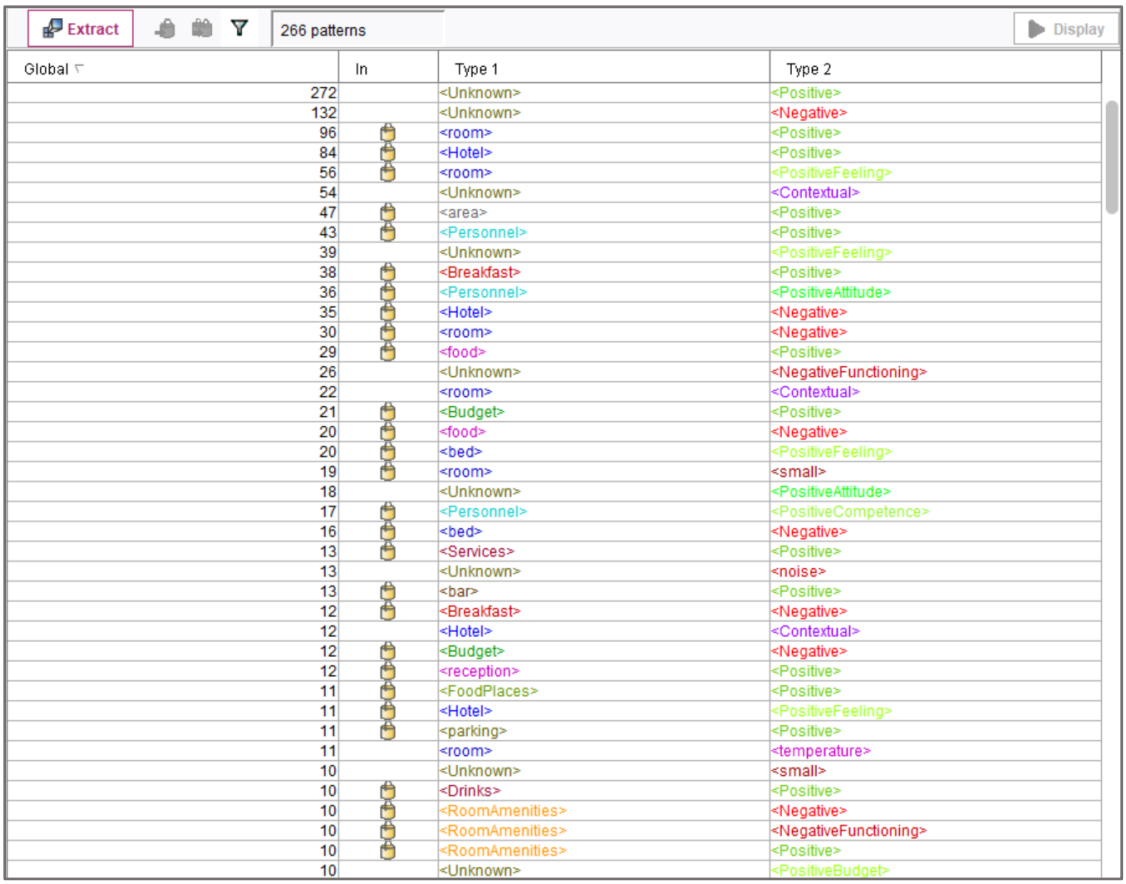


Figure 9.43 Extracted patterns using a TAP file with a wider range of noun subjects defined

We can see how these categories appear in the categories and concepts window. Figure 9.44 shows that we can create a hybrid categorisation scheme based on subject and sentiment (e.g., **Room Positive**) or just subject-based categories (e.g., **Internet**). Finally, Figure 9.44 shows a new category web plot revealing a strong three-way relationship between the categories **Hotel Positive**, **Staff Positive** and **Room Positive**.

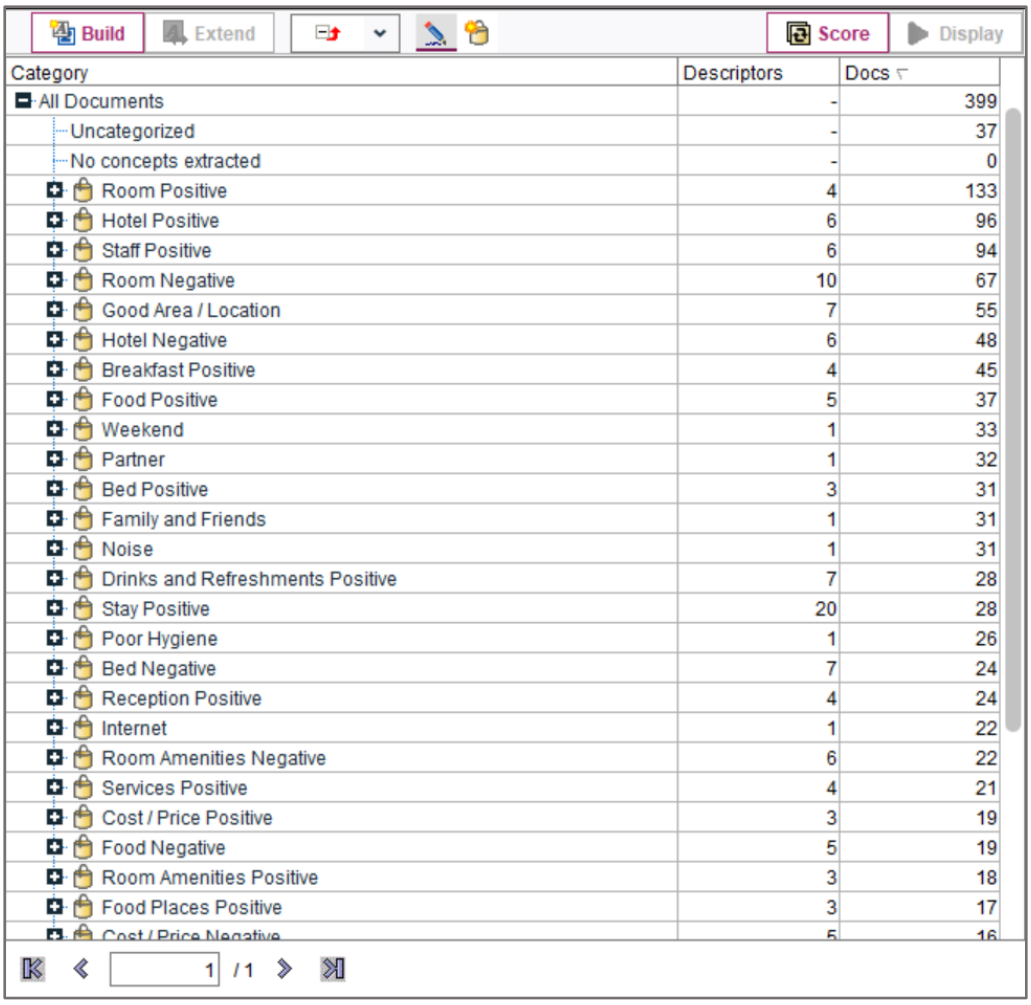


Figure 9.44 Extracted patterns using a TAP file with a wider range of noun subjects defined

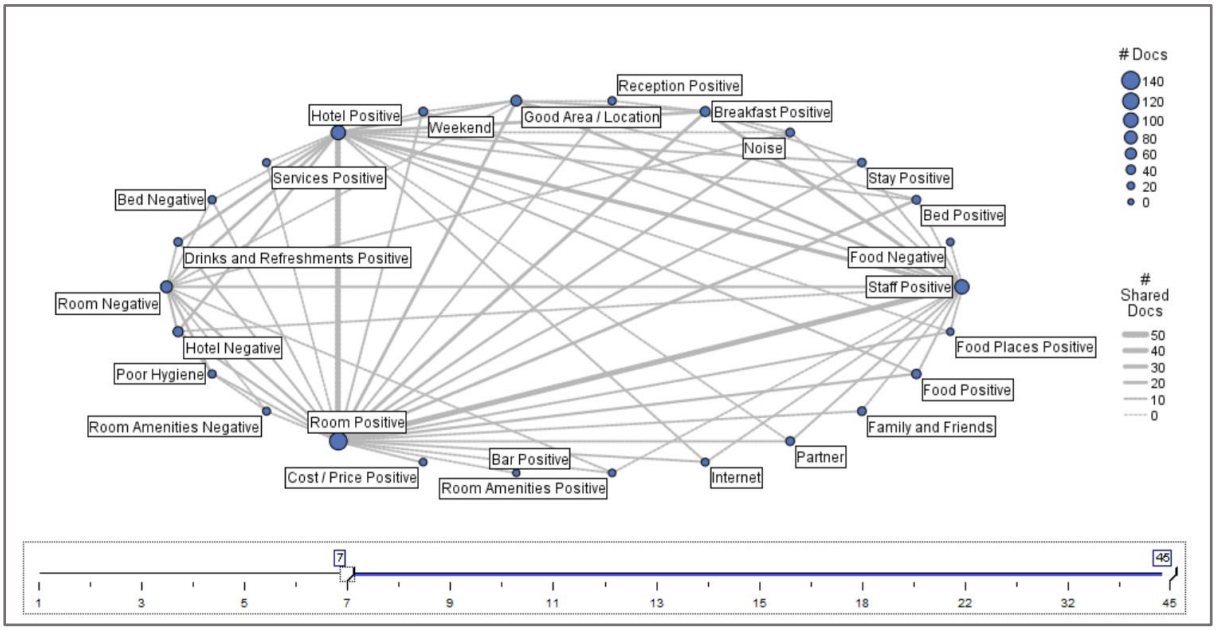


Figure 9.45 Web category plot show relationships between new sentiment and subject-based categories

Practice Exercise – Chapter 9

Within the folder **Student Exercises** open the following stream:

Chapter_09_Practice.str

1. Right-click on the text mining node and load the resource template **Car Rental** that you created earlier. Run the node to extract the concepts.

2. Once the extraction process has finished, go to the **Categories** menu and use the **Import Predefined Categories** procedure to again import the file **Car_Rental_Predefined_Categories.xlsx.**

3. Having imported some categories, we can enrich the existing categorisation by creating some key sentiment-based categories. To do so, switch to the **Text Link Analysis** window and click the **Extract** button. The software will now use the default TLA pattern rules to perform text link analysis on the data set.

4. Once the TLA extraction is complete, you can create categories that capture positive and negative sentiment as it relates to customer service. To do so, sort the column **Type 1** by clicking on the column header. Scroll down until you see the type group **<CustomerSupport>**.

Now use the CTRL-CLICK to select each instance of **<CustomerSupport>** and negative sentiment (i.e., all the types in red beginning with the word **Negative**). When you have finished selecting them, right-click and select **Add to Category** followed by **Create New Category.**

Repeat the process for all the type groups in green beginning with the word **Positive**.

5. Return to Categories and Concepts window and rename the categories to **Negative Customer Service** and **Positive Customer Service** respectively. This approach represents *one* way in which you can easily create sentiment-based categories.

6. A second approach is to define your own custom TLA rules. To do this, switch the window to the **Resource Editor** and click the tab marked **Text Link Rules**. Here you can view all the current TLA Rules. To create our own custom rule, we will first create a macro group. Right-click on the **Macros** folder and **choose Create Macro**. Call the new macro group **mSubjects** and click **Yes** on the pop-up dialog.

7. In the element list below, add the following types in separate rows. You can either right-click and add them from the pop-up list, or type them in, but prefix each one with a **$** character:

- **Customer**
- **Customer Support**
- **Fast**
- **Fuel**
- **Insurance**
- **Personnel**
- **Service**
- **Slow**
- **WaitTime**
- **Staff**
- **Upgrade**

Click, **Apply** to finish.

8. Now we can define a custom rule. Before we so do, first, disable the existing rulesets. Right-click on any existing rulesets in the folder labelled **Rules** and select **Disable**. You will note that these rulesets are greyed out indicating they are no longer enabled.

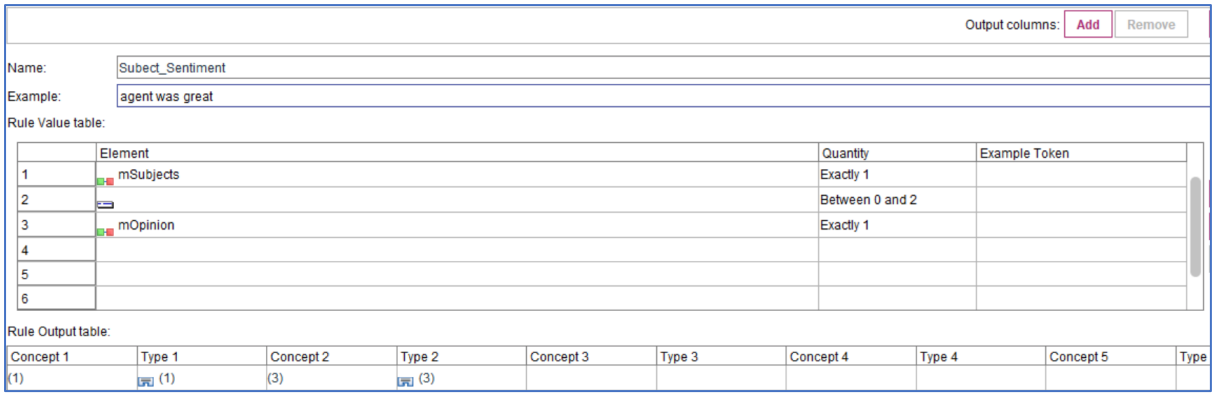
Right-click to create a new ruleset call it **Custom**. Now, right-click within the new ruleset folder and create a new rule. Call it **Subject_Sentiment.**

In the first row of the Element table, add the macro group **mSubjects** (tip: type **$mSubjects** – note it is case sensitive).

In the second row, right-click and select the **<ANY TOKEN>** element. Edit the adjacent **Quantity** cell so that it reads **Between 0 and 2**.

In the third row, add the macro group **mOpinion** (tip: type **$mOpinion**).

In the **Rule** Output table drag **mSubjects** to the first row cell under **Concepts 1** and drag **mOpinion** to the first row cell under **Concepts 2**. The table should look like the following image.



Output columns: Add Remove

Name: Subect_Sentiment

Example: agent was great

Rule Value table:

	Element	Quantity	Example Token
1	mSubjects	Exactly 1	
2		Between 0 and 2	
3	mOpinion	Exactly 1	
4			
5			
6			

Rule Output table:

Concept 1	Type 1	Concept 2	Type 2	Concept 3	Type 3	Concept 4	Type 4	Concept 5	Type
(1)	(1)	(3)	(3)						

9. To test the new rule, return to the **`Text Link Analysis`** window and click **`Extract`**.

If the rule has been correctly defined, you should see a list of new text link patterns. Some could be used to form their own sentiment-based categories whilst others could be added to existing categories.

We could at this stage return to the TLA rule editor and create a reversed version of this rule, so that it matches with occurrences of the sentiment *followed by* the subject rather than the other way round. Feel free to experiment with this or to use these rules to enhance the current categorisation.

Global	In	Type 1	Type 2
24		<CustomerSupport>	<Positive>
7		<Staff>	<Positive>
6		<CustomerSupport>	<PositiveAttitude>
6		<WaitTime>	<Negative>
5		<CustomerSupport>	<Negative>
4		<CustomerSupport>	<Contextual>
4		<CustomerSupport>	<PositiveCompetence>
4		<Personnel>	<Positive>
4		<upgrade>	<PositiveBudget>
3		<Staff>	<Negative>
2		<CustomerSupport>	<NegativeAttitude>
2		<Personnel>	<PositiveAttitude>
2		<Staff>	<PositiveCompetence>
2		<WaitTime>	<slow>
2		<slow>	<Negative>
1		<CustomerSupport>	<NegativeCompetence>
1		<CustomerSupport>	<Uncertain>
1		<CustomerSupport>	<fast>
1		<Customer>	<NegativeAttitude>
1		<Insurance>	<Contextual>
1		<Insurance>	<Positive>

Chapter 10 Managing Resources and Models

10.1 Advanced Resources

We begin this chapter by looking at the various advanced options available in the middle tab of the resource editor window.

10.1.1 Fuzzy Grouping

The first option we encounter in the advanced resources tab, is the exception list that forms the fuzzy grouping control. Earlier, we encountered fuzzy grouping as part of the extraction options in the expert tab of the text mining node, where it is used to help identify and correct misspelled words by temporarily stripping vowels and double or triple consonants from extracted words and then comparing them to a list of candidate target terms. By default, fuzzy grouping is turned off. In the extraction settings expert tab, its application can be limited to words of a minimum size via the **accommodate spelling for a minimum root character limit** setting. Fuzzy grouping however can lead to errors whereby unrecognised words are thought to be misspelled versions of terms in the current dictionary resources. As a result, the exception list allows us to identify word pairs that the fuzzy grouping algorithm should ignore, as they are legitimately separate terms rather than misspellings. In Figure 10.1 we can that the second example, shows the words **hostile** and **hostel** as an exception pair. It's useful to define exception pairs, especially when working with jargonistic or esoteric terms that are unlikely to exist in the standard dictionaries. By using the fuzzy grouping algorithm, you may find that terms are extracted and replaced by other words, on the basis that algorithm thinks they are simply spelling mistakes, before they are added to an inappropriate type group. It's easy to add additional word pairs in this list, as long as we only add one pair per line and separate the two terms with a tab character.

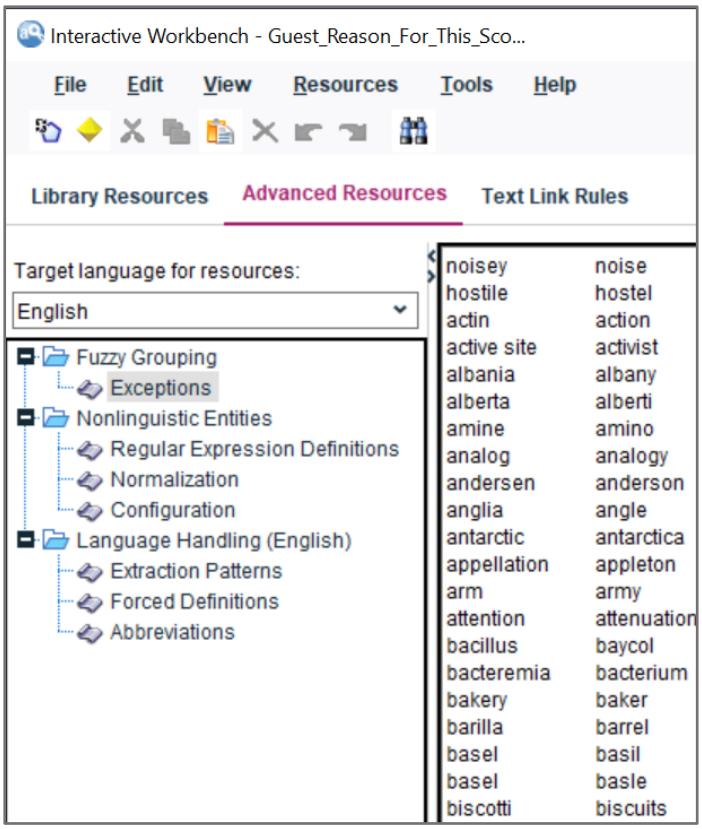


Figure 10.1 Fuzzy grouping exception list in the advanced resources tab

10.1.2 Nonlinguistic Entities

The phrase **nonlinguistic entities** refers to special text terms which aren't words *per se*, but rather convey information such as e-mail address, URLs, phone numbers as well as dates and currencies. In this section of the advanced resources, nonlinguistic entities are defined and controlled using **regular expressions**. Regular expressions are a code standard used to parse and match specific text strings that adhere to particular patterns or syntactic order. Figure 10.2 shows the regular expression code for recognising email addresses. Lines beginning with a **#** character are simply comments used to provide a title or example text for the code snippet. We can break the expression up in the following way:

[email]

- This defines a macro name for the subsequent block of regular expression code.

#@# anything@anything.whatever.etc

- This is text string preceded by a **#** character is simply an example used to illustrate what the code is doing. The code will look for the **@** symbol as part

of a text string containing period symbols, as this indicates the string is most probably an email address.

regexp1=[a-z0-9][a-z0-9._-]+@([a-z0-9_-]+\.)+[a-z0-9]+

- This is the actual regular expression for identifying emails. It is defined within the email code block as **regexp1**, and has been formulated to look for alphanumeric strings (i.e., strings containing numbers and/or letters) using the code **[a-z0-9]**, possibly followed by more alphanumeric strings which might also contain underscores and/or period symbols followed by an **@** symbol and so on.

```
#****************************************************************
#                    EMAIL
#****************************************************************
[email]

#@# anything@anything.whatever.etc
regexp1=[a-z0-9][a-z0-9._-]+@([a-z0-9_-]+\.)+[a-z0-9]+

caseSensitive=0
accentSensitive=0
```

Figure 10.2 Regular expression code for identifying email addresses

Scrolling through the regular expressions definitions, we can see code snippets for identifying amino acids, IP addresses and percentages. Users in certain industries might define their own regular expressions to courier tracking id's, national insurance numbers or disease classification codes. Figure 10.3 shows an example of a regular expression added to the definitions list that identifies UK postcodes.

```
#****************************************************************
#                  UK POSTCODES
#****************************************************************
[Postcode]

#@# EC2M 1QS

regexp= ([Gg][Ii][Rr] 0[Aa]{2})|((([A-Za-z][0-9]{1,2})|(([A-Za-z][A-Ha-hJ-Yj-y][0-9]{1,2})|(([A-Za-z][0-9][A-Za-z])|([A-Za-z][A-Ha-hJ-Yj-y][0-9][A-Za-z]?))))\s?[0-9][A-Za-z]{2})


caseSensitive=0
accentSensitive=0
```

Figure 10.3 Regular expression code for identifying UK postcodes

Beneath the regular expression definition list is the **Normalization** list. When nonlinguistic entities are extracted, the entities themselves are normalized so that they have a consistent format. For example, text strings and currency symbols such

as **euro**, **€, eur** or **eu** are replaced by the prefix **EUR**. Disabling a normalisation entry can be done simply by placing a **#** symbol at the beginning of the respective line. By default, dates in an English template are recognized in the American style *month, day, year* format. If you wish to change this to the UK *day, month, year* format, disable the **format:US** line (i.e., add a **#** character at the beginning of the line) and enable the **format:UK** by removing the **#** character from that line.

```
[english/Currency]

# Dollars
SGD:singaporean dollars:singaporean dollar:singapore dollars:singapore dollar:sg dollars:sg dollar:sgd:s $:s$
AUD:australian dollars:australian dollar:australia dollars:australia dollar:au $:au$:a$:aud
BSD:bahamian dollars:bahamian dollar:bahamas dollars:bahamas dollar:b $:b$:bsd
BBD:barbados dollars:barbados dollar:bds $:bds$:bbd
BZD:belizean dollars: belizean dollar:belize dollars:belize dollar:bz $:bz$:bzd
BMD:bermudan dollars:bermudan dollar:bermuda dollars:bermuda dollar:bmd
BND:brunei dollars:brunei dollar:bnd
CAD:canadian dollars:canadian dollar:canada dollars:canada dollar:c $:c$:cad
KYD:cayman islands dollars:cayman islands dollar:cayman dollars:cayman dollar:kyd
XCD:east caribbean dollars:east caribbean dollar:xcd
FJD:fijian dollars:fijian dollar:fiji dollars:fiji dollar:fjd
GYD:guyanese dollars:guyanese dollar:guyana dollars:guyana dollar:gyd
HKD:hong kong dollars:hong kong dollar:hk dollars:hk dollar:hg dollars:hg dollar:hk $:hk$:hg $:hg$:hkd
JMD:jamaican dollars:jamaican dollar:jamaica dollars:jamaica dollar:jmd
LRD:liberian dollars:liberian dollar:liberia dollars:liberia dollar:lrd
NAD:namibian dollars:namibian dollar:namibia dollars:namibia dollar:nad
TWD:new taiwanese dollars:new taiwanese dollar:new taiwan dollars:new taiwan dollar:ntd:nt $:nt$:nt:twd
NZD:new zealand dollars:new zealand dollar:kiwi dollars:kiwi dollar:nz $:nz$:nzd
SBD:solomon islands dollars:solomon islands dollar:solomon dollars:solomon dollar:sbd
SRD:suriname dollars:suriname dollar:srd
TTD:trinidad and tobago dollars:trinidad and tobago dollar:t&t dollars:t&t dollar:ttd
ZWD:zimbabwean dollars:zimbabwean dollar:zimbabwe dollars:zimbabwe dollar:z $:z$:zwd
USD:united states dollars:united states dollar:u.s. dollars:u.s. dollar:us dollars:us dollar:dollars:dollar:usd:$ us:$us:us $:us$:$
```

Figure 10.4 Normalisation expressions of various currency formats for English language templates

The **configuration** section allows users to enable and disable the nonlinguistic entity types from the extraction process. Disabling unneeded nonlinguistic entities has the effect of speeding up the processing time. The configuration list has three elements:

- **#name** – the case sensitive macro identifier of the nonlinguistic entity as defined in the regular expressions definitions list.
- **Language** – The document language code. The codes include **0** = any language which is normally used whenever a regular expression is not language dependent, such as IP addresses or URLs. Other codes include **1** = French; **2** = English; **4** = German; **5** = Spanish; **6** = Dutch; **8** = Portuguese; **10** = Italian.
- **Code** – The code here refers to the parts of speech role. Most entities use the value **s** where **s** = stop word. This means the entity will be extracted on its own. Other codes however are **a** = adjective and **n** = noun. Nonlinguistic entities are extracted before any parts of speech extraction patterns are applied to identify the word's potential role in a phrase. As an example,

percentages are coded as **a**. So, when the string **15%** is extracted as a nonlinguistic entity, it will be assigned an adjective role in the extraction patterns parts of speech algorithm. One of these pattern rules extracts phrases on the basis that they are comprised of an adjective followed by two nouns (the parts of speech code for this is **ann**). On this basis, a phrase like **15% sales tax** would be extracted rather than just the entity **15%** on its own.

10.1.3 Language handling

In the **Language Handling** section, users can edit extraction patterns, force definitions for those patterns, and declare abbreviations for the relevant selected language. Although, the subject of parts of speech patterns and their usage in deciding which words and phrases are extracted was introduced earlier in this course, here we finally encounter the area where parts of speech patterns are defined under the **extraction patterns** list.

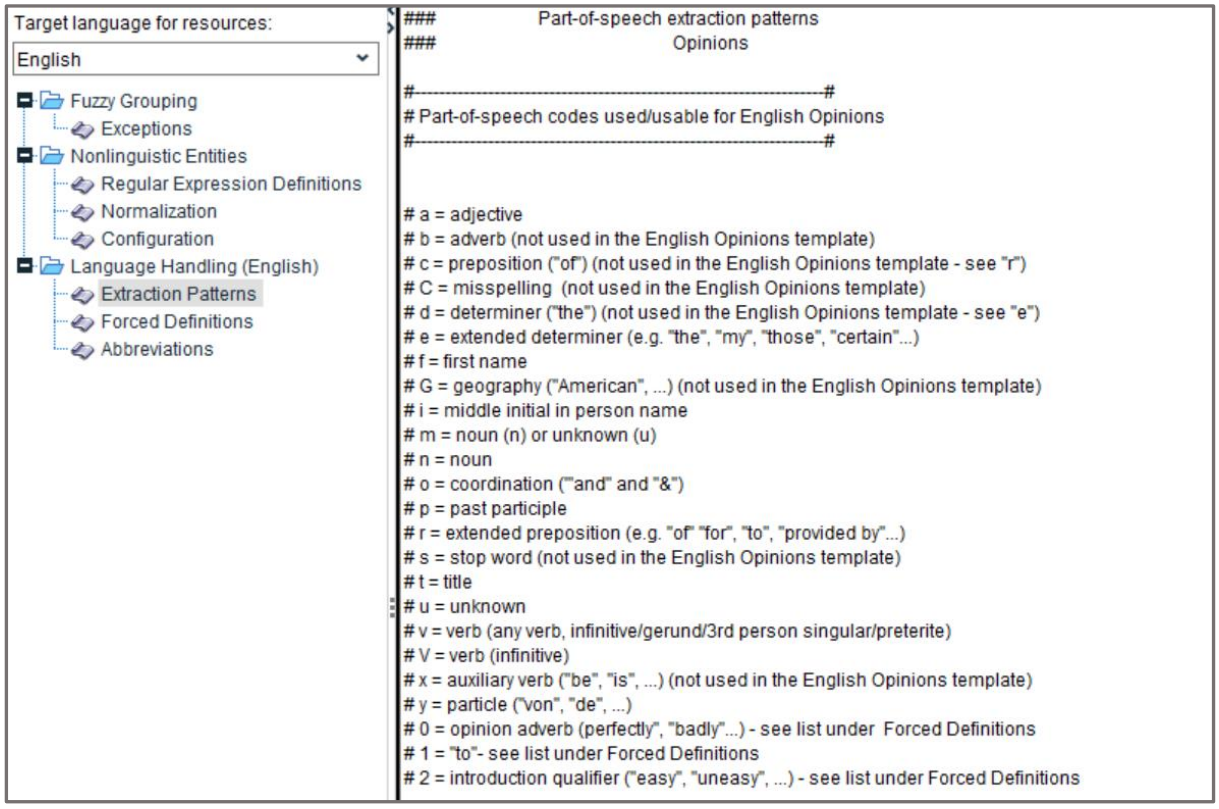


Figure 10.5 Language handling section showing the extraction patterns list for editing parts of speech

The first section in the list provides a useful key telling us the character codes for each part of speech. The following is an abbreviated list of the codes used when working with the English Opinions template (the Basic Resources template contains a slightly shorter range of codes).

- a = adjective
- b = adverb *
- c = preposition (e.g., **of**) *
- C = misspelling *
- d = determiner (e.g., **the**) *
- e = extended determiner (e.g., **the**, **my**, **those**, **certain**)
- f = first name
- G = geography (e.g., **American**) *
- i = middle initial in person name
- m = noun (n) or unknown (u)
- n = noun
- o = coordination (e.g., **and**, **&)**
- p = past participle
- r = extended preposition (e.g., **of**, **for**, **to**, **provided by**)
- s = stop word*
- t = title
- u = unknown
- v = verb (any verb, infinitive/gerund/3rd person singular)
- V = verb (infinitive)
- x = auxiliary verb (e.g., **be**, **is**) *

* indicates that this code is not used in the English Opinions template

Some examples of parts of speech patterns are as follows:

Parts of speech	Pattern	Example
Verb-Noun	**Vm**	maintain contact
Noun-Noun-Extended Preposition-Noun	**mmrm**	patient care for children
Adjective-Noun-Noun	**amm**	fast reaction time
Noun-Extended Preposition-Adjective-Noun	**mram**	food for hungry guests
Adjective-Coordination-Adjective-Noun	**aoam**	medical and legal assistance

Figure 10.6 Example parts of speech patterns

Most parts of speech tend to be centred around the use of nouns. Verbs are not extracted on their own. Figure 10.7 shows how easy it is to request that terms that are detected as verbs are included in the extraction process: we can see that a lowercase **v** has been added to the extraction patterns.

```
# Unsused Pos codes
# g, h, j, k, l, q, w, z, all upper-case letters except "C", "G" and "V", all digits except 0, 1 and 2

# speak
V
#                able to purchase
1V
```

Figure 10.7 Editing the parts of speech pattern list so that single term verbs are extracted

Figure 10.8 shows the effects of editing the parts of speech pattern list in this way. We can see that a number of single verb terms such as **get**, **booked**, **do** and **say** are now included in the extraction results. Of course, these words do not occur in any compiled dictionaries so are all typed as **unknown**.

	Concept	In	Global	Docs	Type (Selected)
1	stay	fx	90	76 (19%)	<Unknown>
2	night	fx	79	60 (15%)	<Unknown>
3	get		41	37 (9%)	<Unknown>
4	booked		38	34 (9%)	<Unknown>
5	do		33	30 (8%)	<Unknown>
6	say		25	23 (6%)	<Unknown>
7	place		25	23 (6%)	<Unknown>
8	bath		24	24 (6%)	<Unknown>
9	experience		22	22 (6%)	<Unknown>
10	made		21	21 (5%)	<Unknown>
11	got		20	19 (5%)	<Unknown>
12	floor		20	17 (4%)	<Unknown>
13	full		20	18 (5%)	<Unknown>
14	time	fx	20	20 (5%)	<Unknown>
15	staying		20	20 (5%)	<Unknown>
16	told		20	16 (4%)	<Unknown>
17	think		19	18 (5%)	<Unknown>

Figure 10.8 Single term verbs extracted as a result of editing the parts of speech pattern list

Often, we find that a particular word could be viewed as a noun, adjective or verb depending on how it's used in a sentence (e.g., **clean**). The **Forced Definitions** section in the language handling options allows us to force a word to take a particular grammatical meaning. Alternatively, by marking a word as a **stop word** with a lowercase **s** code, we can prevent it from being extracted into compound words or phrases. However, if a word match is explicitly declared as a term in a compiled dictionary, it will still be extracted. One of the phrases in the forced definitions list of the Opinions template is **easy to**. Within the Basic Resources template, as there are no forced definitions, and the word **easy** does not appear in the compiled dictionaries, the phrase **easy to** is not extracted. As a result, when applying the Opinions template to a sentence like **I found it easy to work with**, results in the concept **easy to work** being extracted, whereas using the Basic Resources template, results in just the noun **work** being extracted.

The last part of this section controls abbreviations. When the extractor engine is processing text, in most cases it regards any period character encountered as an

indication that a sentence has ended. This of course is usually correct, the exception being when the period follows an abbreviation. An extensive list of commonly used abbreviations are already in the compiled resources for each language. For English resources, these abbreviations include terms like **etc.**, **dept.** and titles preceding personal names (such as **Mr.**, **Mrs.** or **Dr.**). Abbreviations that appear without the use of a period character aren't affected by these definitions. If a user's text data contains abbreviations that have been misunderstood by the system, they can add additional abbreviations to this section. This will not be necessary however, if the abbreviation is already defined as a term in a type dictionary or as a synonym.

10.2 Managing Resources

We've already seen how many of the resources for a project in Modeler Text Analytics are stored in collections of libraries. Moreover, these libraries are in turn comprised of individual dictionary files such as type dictionaries, synonym/substitution dictionaries and dictionaries of excluded terms.

In this way, a collection of libraries can be brought together to form a resource template that in turn can be applied to a specific application, area of study or industry domain. Moreover, these resource templates may include hidden compiled libraries that identify entities such as people, geographical places and organizations, and special advanced resources that control things like parts of speech extraction patterns, regular expressions and exception lists to prevent errors caused by the spell-checking algorithm.

Thus far, we have viewed and edited these resources using the resource editor within a particular stream-based interactive session. However, IBM SPSS Modeler also allows us to edit resource files independently of a specific stream or interactive session using the **Template Editor**. To view this editor, from the main menu within IBM SPSS Modeler, click:

Tools

Text Analytics Template Editor

Figure 10.9 shows the menu location for the editor.

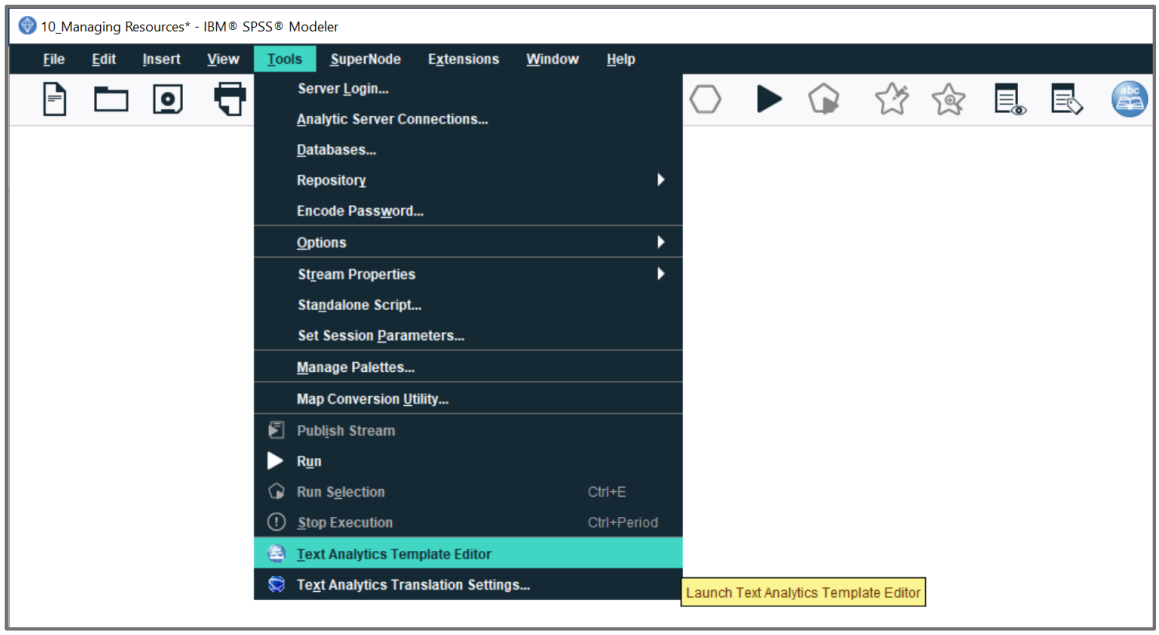
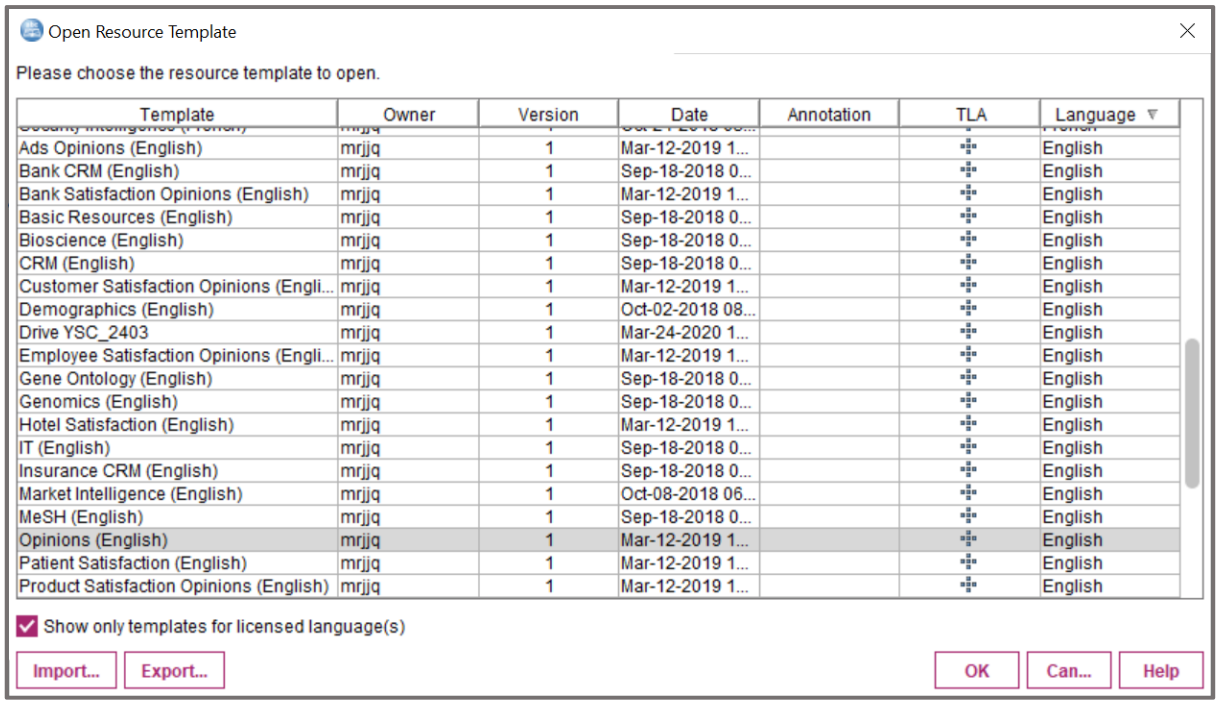


Figure 10.9 Accessing the 'Text Analytics Template Editor' from IBM SPSS Modeler

The Open Resource Template dialog is displayed as shown in Figure 10.10. We can view information related to when the template was created, its associated language and whether it contains TLA patterns.



Template	Owner	Version	Date	Annotation	TLA	Language
Ads Opinions (English)	mrjjq	1	Mar-12-2019 1...			English
Bank CRM (English)	mrjjq	1	Sep-18-2018 0...			English
Bank Satisfaction Opinions (English)	mrjjq	1	Mar-12-2019 1...			English
Basic Resources (English)	mrjjq	1	Sep-18-2018 0...			English
Bioscience (English)	mrjjq	1	Sep-18-2018 0...			English
CRM (English)	mrjjq	1	Sep-18-2018 0...			English
Customer Satisfaction Opinions (Engli...	mrjjq	1	Mar-12-2019 1...			English
Demographics (English)	mrjjq	1	Oct-02-2018 08...			English
Drive YSC_2403	mrjjq	1	Mar-24-2020 1...			English
Employee Satisfaction Opinions (Engli...	mrjjq	1	Mar-12-2019 1...			English
Gene Ontology (English)	mrjjq	1	Sep-18-2018 0...			English
Genomics (English)	mrjjq	1	Sep-18-2018 0...			English
Hotel Satisfaction (English)	mrjjq	1	Mar-12-2019 1...			English
IT (English)	mrjjq	1	Sep-18-2018 0...			English
Insurance CRM (English)	mrjjq	1	Sep-18-2018 0...			English
Market Intelligence (English)	mrjjq	1	Oct-08-2018 06...			English
MeSH (English)	mrjjq	1	Sep-18-2018 0...			English
Opinions (English)	mrjjq	1	Mar-12-2019 1...			English
Patient Satisfaction (English)	mrjjq	1	Mar-12-2019 1...			English
Product Satisfaction Opinions (English)	mrjjq	1	Mar-12-2019 1...			English

Figure 10.10 The 'Open Resource Template' dialog

In this example, we will choose the Opinions (English) resource template. To do so, select:

Opinions (English)

OK

In this particular instance, a dialog appears showing that the published version of the Opinions library within this resource template is more recent than the original version. This is due to changes we made earlier in the course. To update the library with the published version simply click:

Update

Figure 10.11 shows this process.

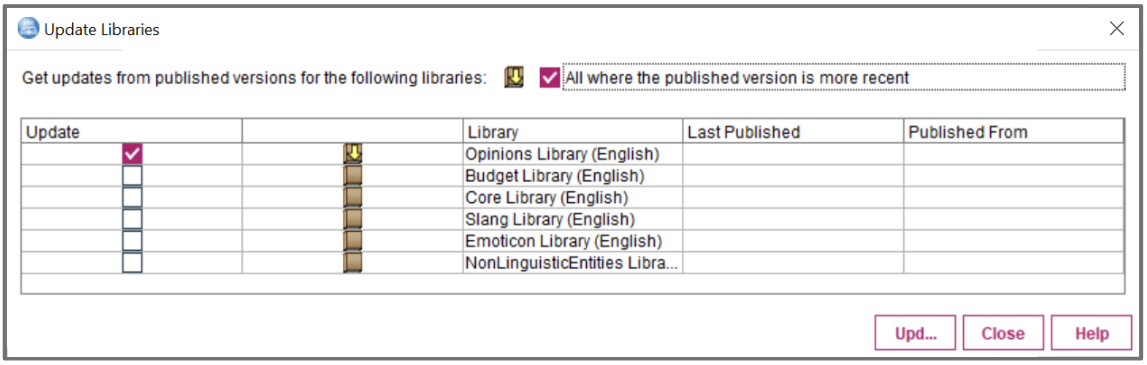


Figure 10.11 The 'Update Libraries' dialog

The resultant text analytics template editor looks identical to the resource editor window that we have become familiar with in the interactive sessions. However, you will notice that there is no way to switch to the categories and concepts window, or the text link analysis window, as the purpose of this editor is simply to edit the resources rather than to extract concepts, analyse patterns or categorise responses.

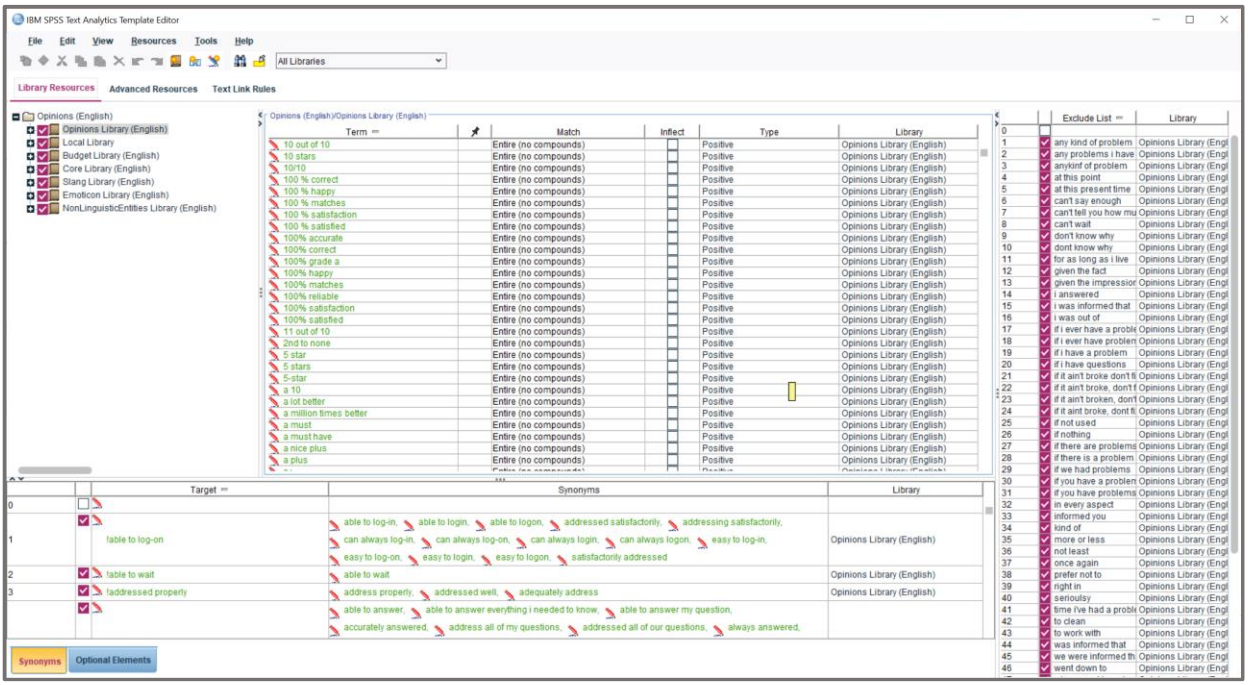


Figure 10.12 The 'Text Analytics Template Editor'

In this case, we don't need to make any changes, so can simply click:

File

Close

Instead, we can make a new resource template by running the text mining node in the following stream:

10_Managing_Resources.str

After the extraction process has completed, we may once again switch to the resource editor within the interactive session. To create a new template based on these edited resources, from the main menu click:

Resources

Make Resource Template

When the **Make Resource Template** dialog is created, specify the following name for the template:

Hotel Reviews

Figure 10.13 shows the dialog at this stage.

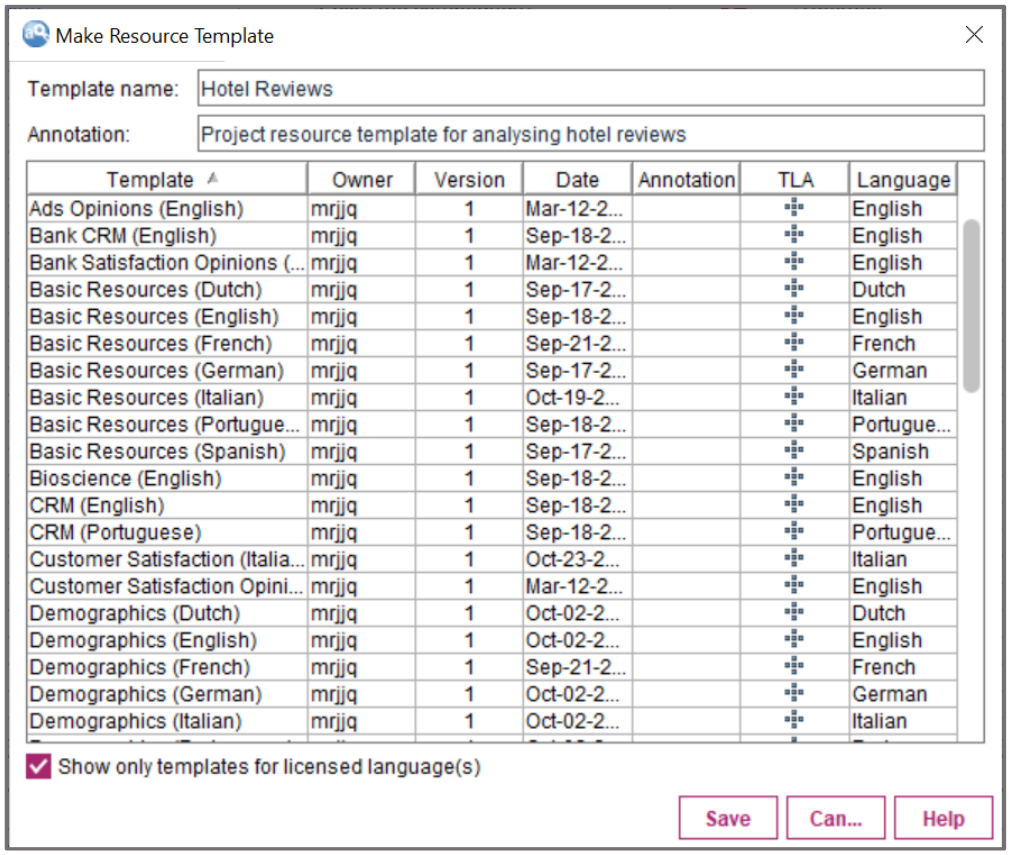


Figure 10.13 Creating a new resource template

We need to bear in mind, that even though the name of the template within the interactive editor has changed, we haven't overwritten the Opinions template that was originally used here. Any template loaded in an interactive session is a *copy* of those resources, for that specific instance of the node. Moreover, the node was already using a template contained within a TAP (text analytics package) file, so if we re-ran the node, it would open up the old resource template saved in that file. If we wanted the TAP file to take account of the new template name, we would need to update it explicitly.

To illustrate how important the difference is between a template resource that is saved and one that has been loaded let's open the stream:

10_Saved_vs_Loaded.str

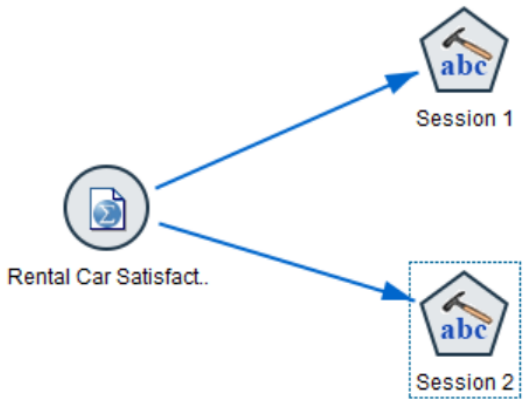


Figure 10.14 The Modeler stream '10_Saved_vs_Loaded.str' with two identical nodes

Here we have data related to customer satisfaction with car rentals. If we execute both text mining nodes, we will generate two identical sessions. But what happens if we make changes in only one of the two sessions?

The extraction results in both sessions show that that the term **avis** is not recognised as an organisation.

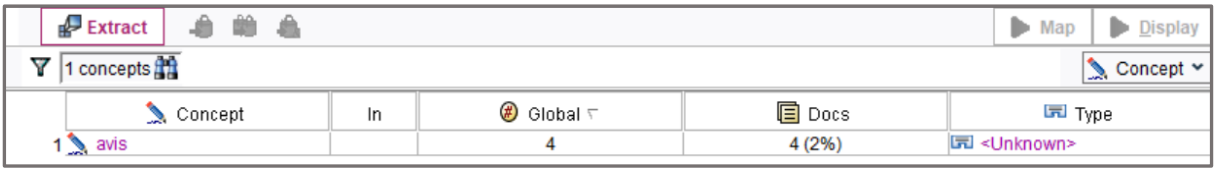


Figure 10.15 Extraction results with unrecognised term 'avis'

Of course, we can easily fix this in resource editor by adding **avis** to the **organisation** type in the **core** library as shown in Figure 10.16.

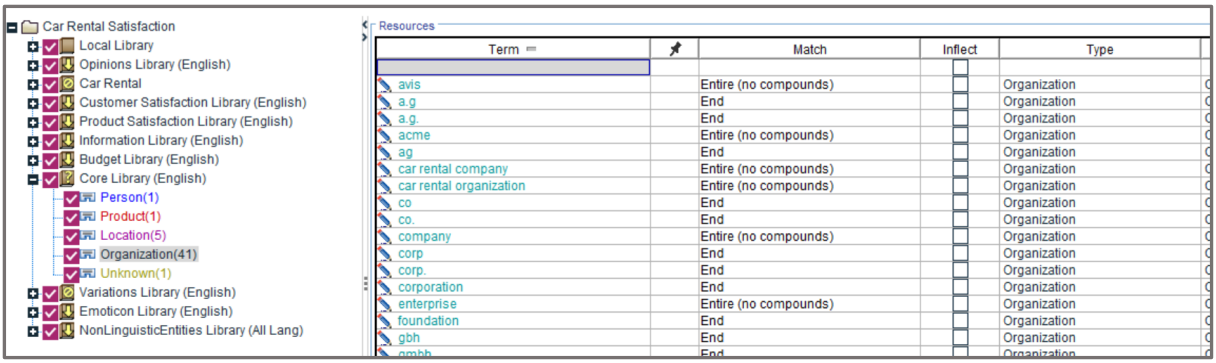


Figure 10.16 The term 'avis' added to the 'organisation' type

If we now re-extract the data in this session we can see that the term is now recognised as an organisation.

Figure 10.17 Re-extracted results with showing the term 'avis' correctly typed as an organisation

Having confirmed this we can now simply overwrite the resource template that was loaded in the text mining node. From the main menu in the resource editor click:

Resources

Make Resource Template

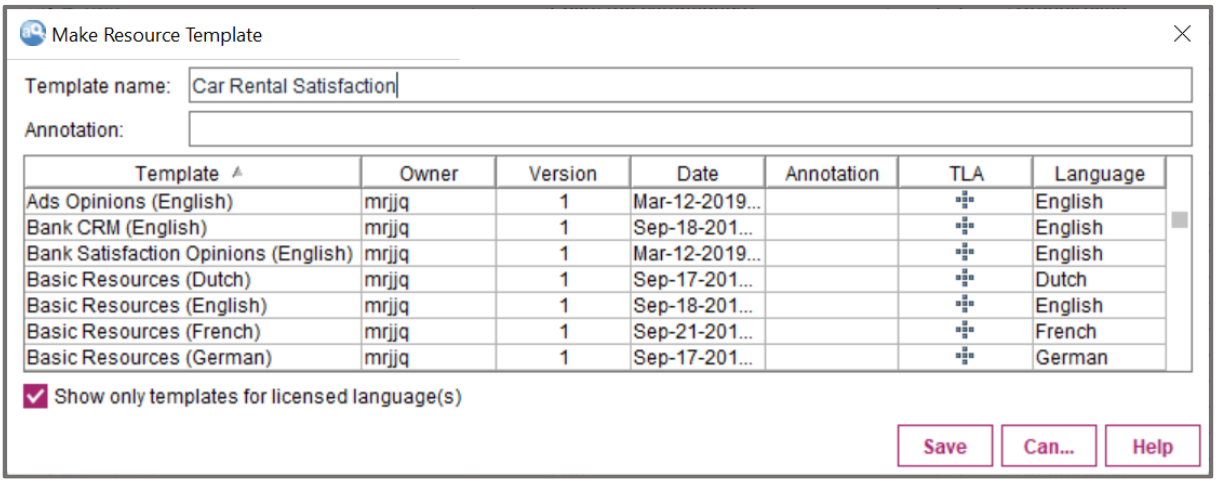


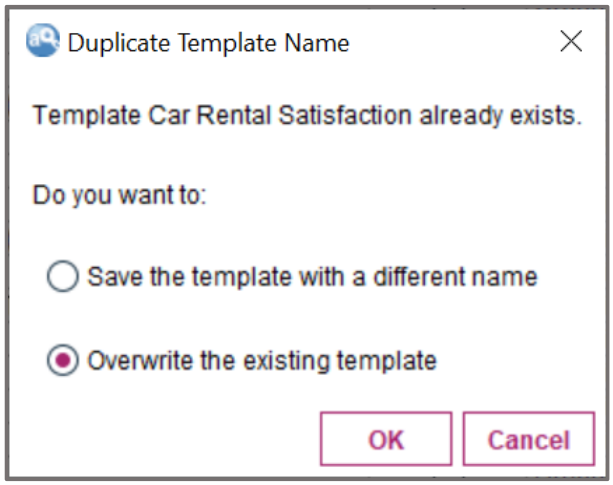
Figure 10.18 Saving an existing resource template

From the pop-up dialog choose:

Save

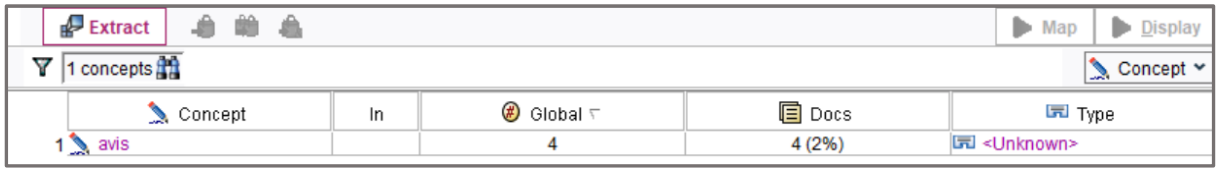
As the dialog in Figure 10.19 shows we can continue without creating another copy of the template by choosing the option:

Overwrite the existing template



10.19 Overwriting the existing template

However, despite the fact that we have updated and overwritten the resource template, when we switch to the second session and force the concepts to be re-extracted, the term **avis** remains in the **unknown** type group.



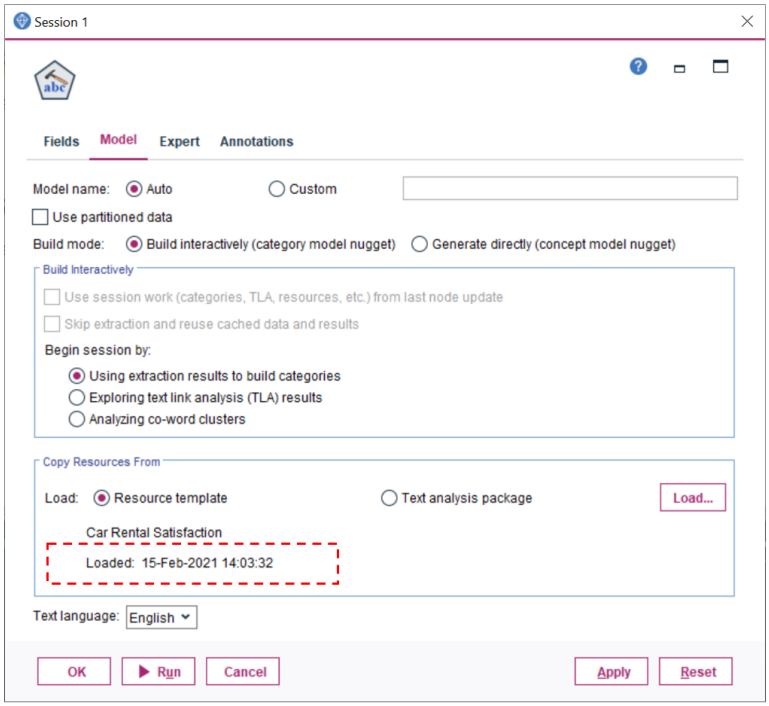
10.20 The term 'avis' remains unrecognised in the second session despite the resource template being updated and saved

This is because the second session is working with a loaded *copy* of the original resource template. So a change made by one session is not reflected in the second, as they are both working with their own loaded copies of the template.

To further illustrate this:

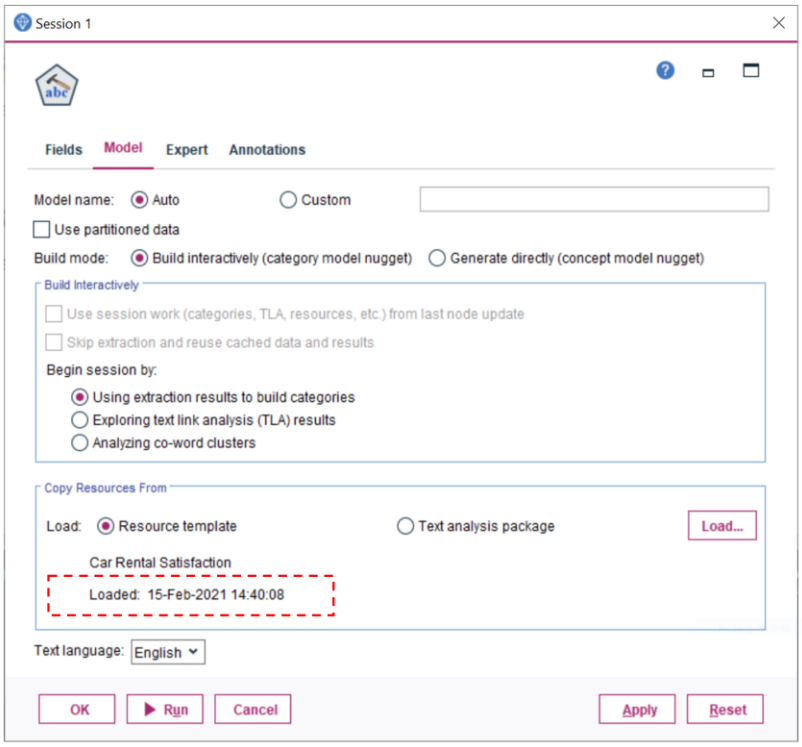
Exit from both sessions without saving any work

Even when we re-execute the *first* session's node, and search for the term **avis** we find that once again it has been typed as **unknown**. We can see why this has occurred when we look at the Model tab of the text mining node. It shows the date and time that the copy of the resource template was loaded and cached in the node. The time stamp will show that this copy **precedes** the edit made to the resources.



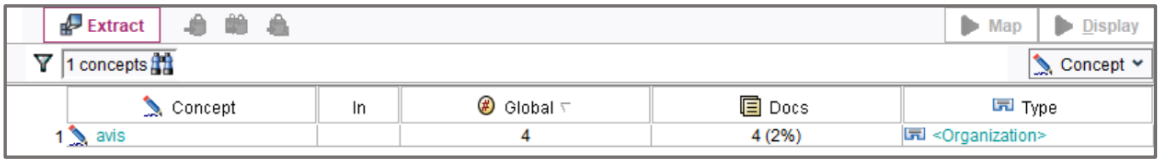
10.20 The text mining node showing it is using a copy of the 'Car Rental Satisfaction' template loaded at 14:03.

If we re-load the template as shown in Figure 10:21, we will now be loading a copy of the *latest* version of the resource template.



10.21 The text mining node showing it is using a copy of the 'Car Rental Satisfaction' template loaded at 14:40.

Now, as Figure 10.22 shows, the term **avis** is correctly identified as an organisation.



10.22 The term 'avis' correctly typed as an organisation

The valuable lesson that this exercise teaches us, is that if we are returning to work on resources that have been updated, we must make sure that we're using the latest version of those updated resources in order to see the effects of any changes made.

10.2.1 Resource file types

So far in this course we've seen that individual published library files can be exported and imported as **.lib** files. We've also seen that we can export and import entire collections of resources including any categories or TLA patterns as text analytics packages (or **.tap**) files. The interactive resource editor also allows us to export resource templates containing multiple libraries as individual **.lrt** files. To demonstrate, within the resource editor click:

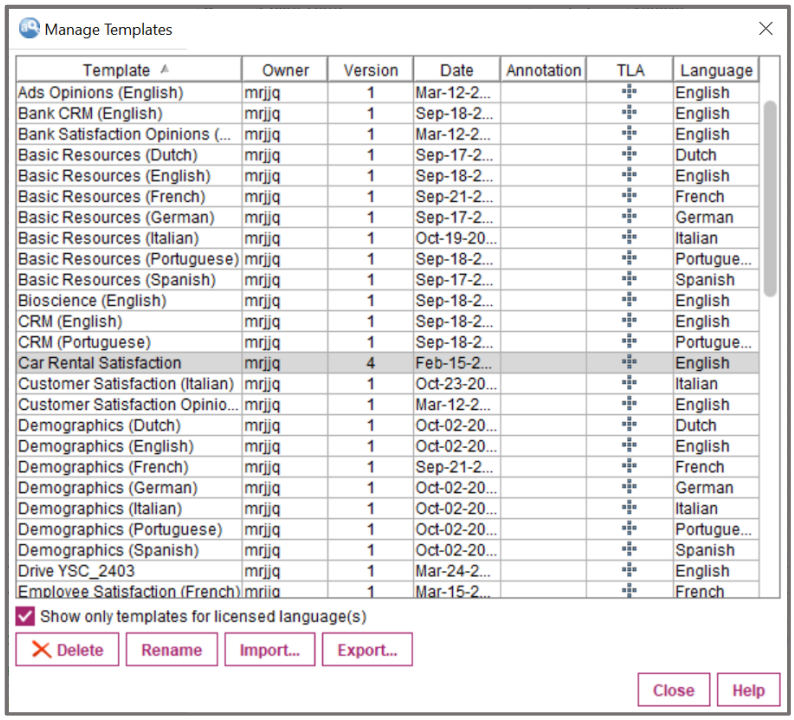
Resources

Manage Resource Templates

Choose the template:

Car Rental Satisfaction

Figure 10.23 shows the manage templates dialog at this stage:

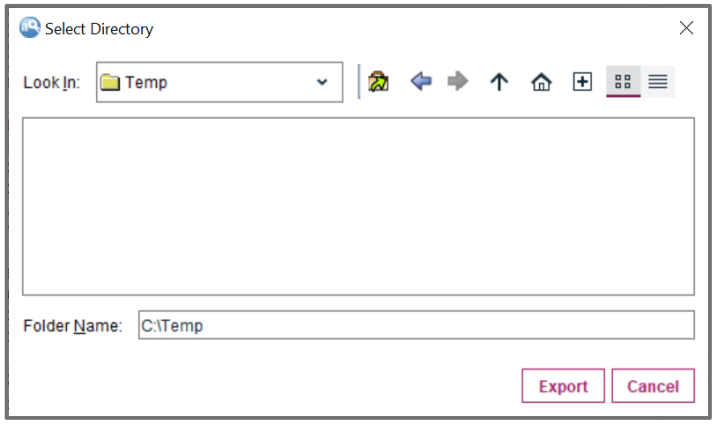


10.23 The 'Manage Templates' dialog

Click the button marked:

Export

Figure 10.24 shows the dialog at this stage.

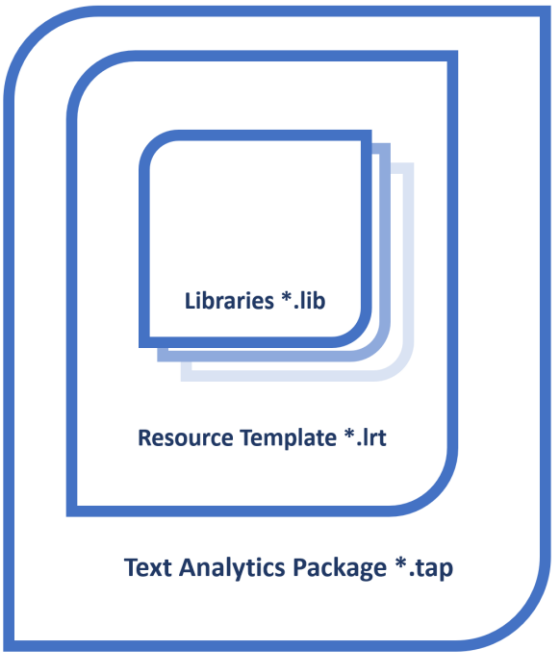


10.24 Exporting a resource template as the file 'Car Rental Satisfaction.lrt'

Choose a folder to export the file and click:

Export

The resource template file is exported. Figure 10.25 reminds us of the relationship between libraries, resource templates and text analytics packages. Remember TAP files contain not just a resource templates with multiple libraries, but also response categories and TLA patterns.



10.25 Relationship between the library, resource template and text analytics package file types

Lastly, we can demonstrate that Modeler Text Analytics contains functionality to backup and restore all our published resources. Backing up resources is a sensible procedure for any collection of edited files, but in the case of software like Modeler Text Analytics it makes sense in case you ever need to deinstall the software and reinstall it later. This is because a reinstall will overwrite any existing resources. In fact, the existing resources are *already* stored in an internal instance of a MySQL database. The backup procedure writes out these resources as a single file with the extension **.tmb**. To illustrate this, within the resource editor, click:

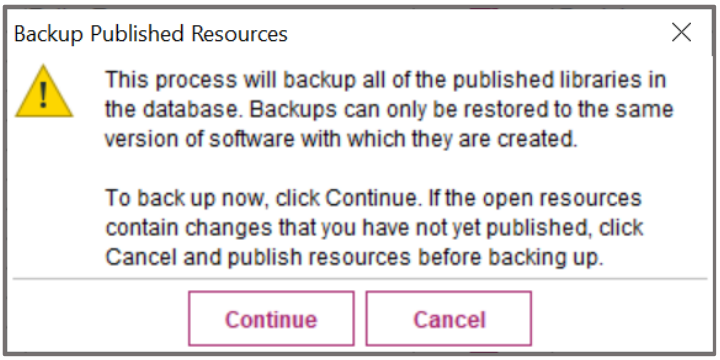
Resources

Backup Tools

Backup Resources

At this point the warning message shown in Figure 10.26 appears telling us that:

- Backups can only be restored to the same version of the software
- Any unpublished resources will not be included in the backup

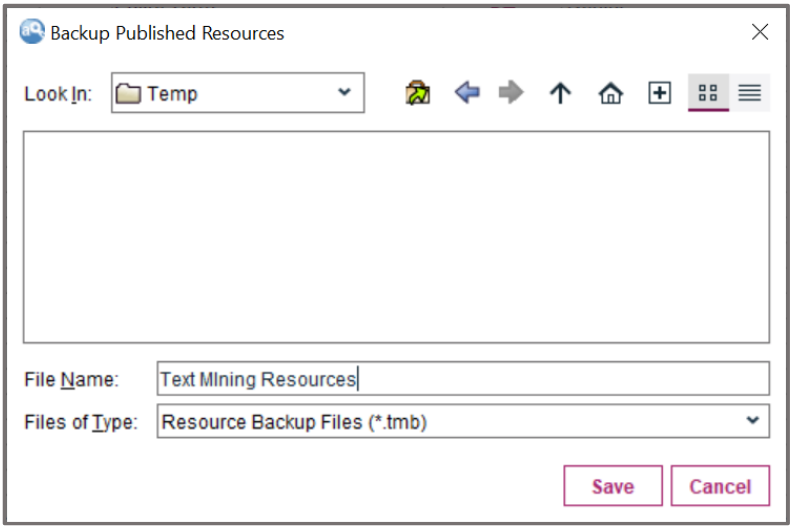


10.26 Backup warning message

To continue click:

Continue

Specify a folder and appropriate file name as shown in Figure 10.27.

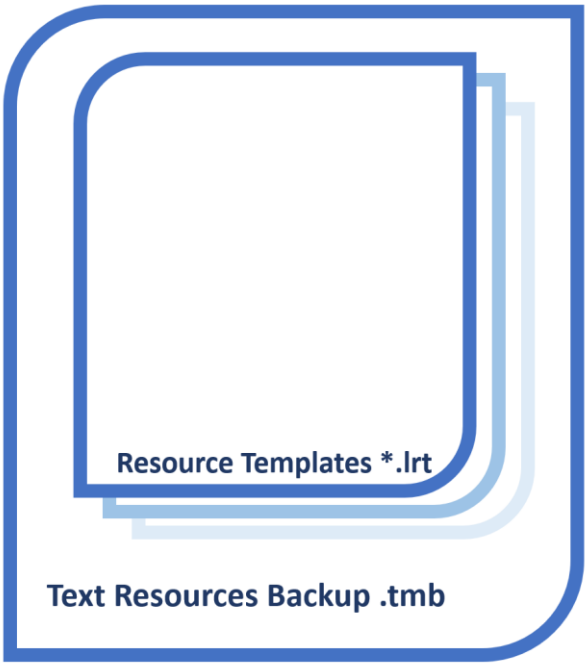


10.27 Completing the backup process: specifying a folder and file name

To finish, click:

Save

Figure 10.28 simply reminds us that unlike **.tap** files, backup **.tmb** files contain the *multiple* resource templates and their respective libraries.



10.28 Relationship between backup *.tmb file types and resource template *.lrt file types

10.3 Working with Text Analytics Models

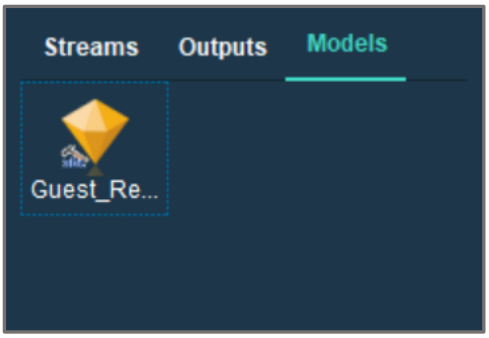
Usually, the ultimate goal of the text mining process is to create a model that enables the accurate classification of text data. Fortunately, once the categorisation process has been completed, this is a simple procedure. Returning to the hotel reviews data in the stream **10_Managing_Resources.str**, after the extraction process has ended, within the categories and concepts window, we can generate a model by simply by clicking:

Generate

Generate Model

Or by clicking the **Generate Model** icon on the main toolbar:

Returning to the stream canvas, we can see that in the Models tab on the right-hand side of the screen, a model nugget has been added.



10.29 Model nugget added to Model tab

We can add the model nugget to the existing stream simply by double-clicking it. To browse the model, we can either:

Right-click on the model nugget on the stream canvas and choose **Edit**

Or

Double-click the model nugget on the stream canvas

Doing this opens up a dialog, separated by tabs, that reveals the model contents and controls how it is applied to text data. Figure 10.30 shows the model tab which allows us to browse the various descriptors in each category. Remember that descriptors can consist of terms, category rules, extracted concepts, types or TLA rules. The check boxes on the left-hand side of the dialog also allow us to choose whether or not these categories should be included in the scoring process.

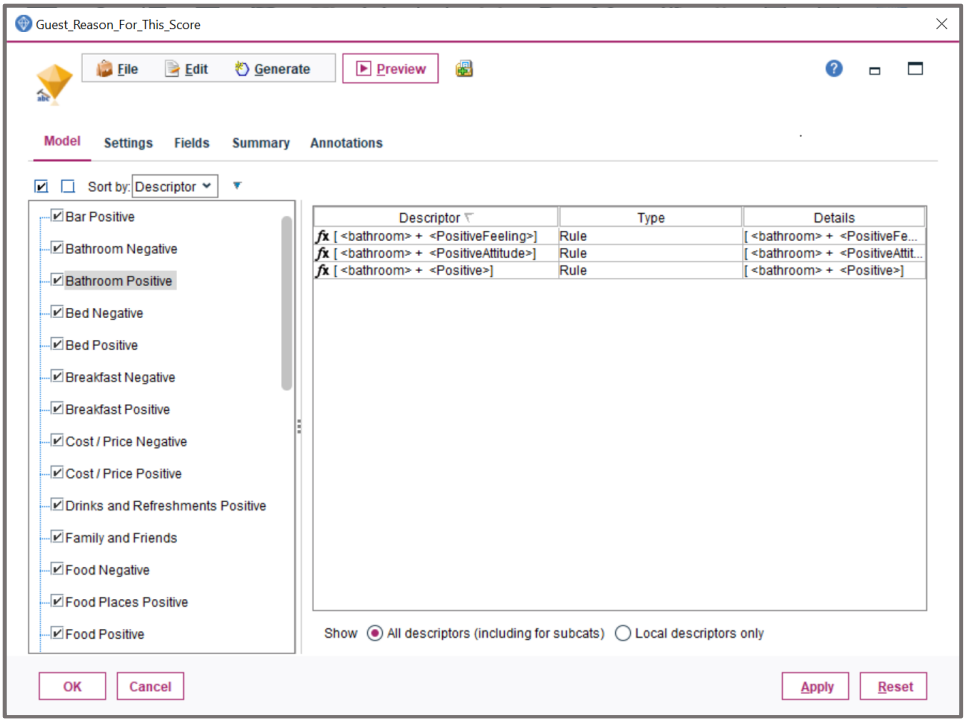


Figure 10.30 Model tab of a text mining model nugget showing categories and their respective descriptors

Clicking the **Settings** tab, reveals a set of controls over how the model scores the text data in order to generate physical fields indicating which individual categories matched the accompanying text. Figure 10.31 shows an annotated image if this tab.

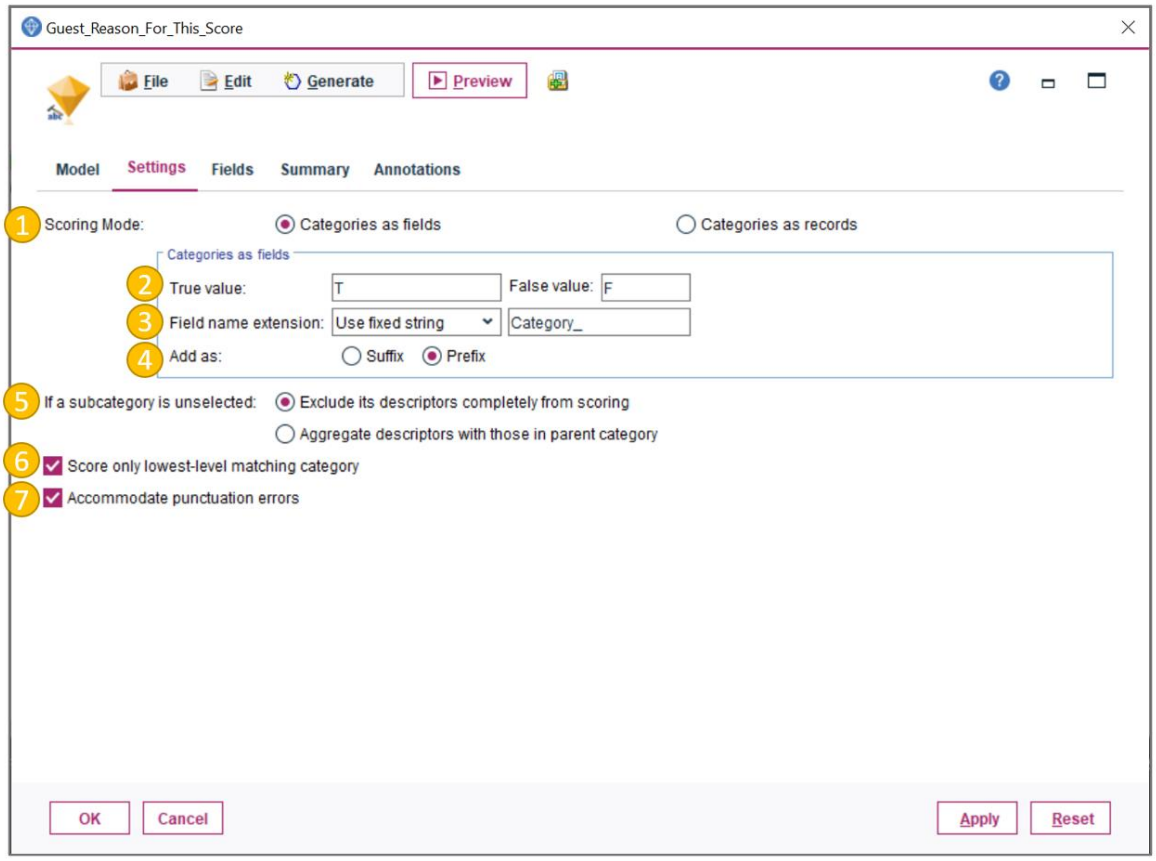


Figure 10.31 Settings tab of the text mining model nugget

The numbered options in Figure 10.31 relate to the following controls:

1. **Scoring Mode** -This option controls how the model generates new fields:
    - **Categories as fields** – this default method creates a series of binary fields that simply indicate whether a record matched or did not match one of the categories in the text mining model. In our case, the model based used on the hotel reviews data created 34 categories, so 34 new fields will be added to each response row in the dataset.
    - **Categories as records** – instead of creating new fields for each category, this method creates a single text category variable but adds new *records* for each category *that the response matches with*. In other words, if the data only matches with 4 out of 34 categories, then only 4 rows of data are created. Rather than having a binary value in each row of the category variable, the values are the actual category labels using in the classification schema. Choosing this option will open a number of sub-options that relate to how the user wants to deal with hierarchical categories.

2. **True value** – This option only relates to the **categories as fields** method. It controls how the user wants the values in the binary variables to indicate that a match has or has not occurred. By default, the value **T** (for **true**) indicates a successful match, and the value **F** (for **false**) indicates no match. There are times when users may prefer to use to numeric values such as **1** and **0** if they wish to perform certain further statistical analyses (such as correlations) that are unable to cope with string values.
3. **Field name extension** - This option only relates to the **categories as fields** method. Each new binary field is given a prefix or suffix label that indicates the variable came from a text categorisation model. This label can either be a fixed string or a category code. The default fixed string label is **Category_**, so a model containing a category called **Noise** would appear as the binary field **Category_Noise**. Users often edit this label to shorten the resulting field name.
4. **Add as** - This option only relates to the **categories as fields** method. It controls whether the field name extension label appears as a **prefix** or a **suffix** e.g., **Category_Noise** vs **Noise_Category**.
5. **If a subcategory is unselected -** This option allows users to specify how the descriptors belonging to subcategories, that *were not* selected for scoring, will be handled. There are two options:
    - **Exclude its descriptors completely from scoring** will cause the descriptors of subcategories that do not have checkmarks (unselected) to be ignored and unused during scoring.
    - **Aggregate descriptors with those in parent category** will cause the descriptors of subcategories that do not have checkmarks (unselected) to be used as descriptors for the parent category. If many levels of subcategories are unselected, the descriptors will be aggregated under the first available parent category.
6. **Score only the lowest-level matching category** – This again relates to hierarchical categories. If the category **Bathroom** contains a subcategory **Shower**, then it is normally shown as **Bathroom/Shower**. Switching off this default option means that if a response mentions the word **shower**, then the category **Bathroom** *and* the category **Bathroom/Shower** are matched. Leaving the default on means that only the category **Bathroom/Shower** is matched.
7. **Accommodate punctuation errors** - This option applies an algorithm that temporarily normalizes text that may contain punctuation errors. This option is regarded as useful when the text is deliberately curtailed or contains the kinds of abbreviations typically encountered in call centre notes, as well as sales management or CRM systems.

We can see the default behaviour of a text mining model by connecting our model nugget to the hotel reviews data source and adding a table node downstream of the model as shown in Figure 10.32.

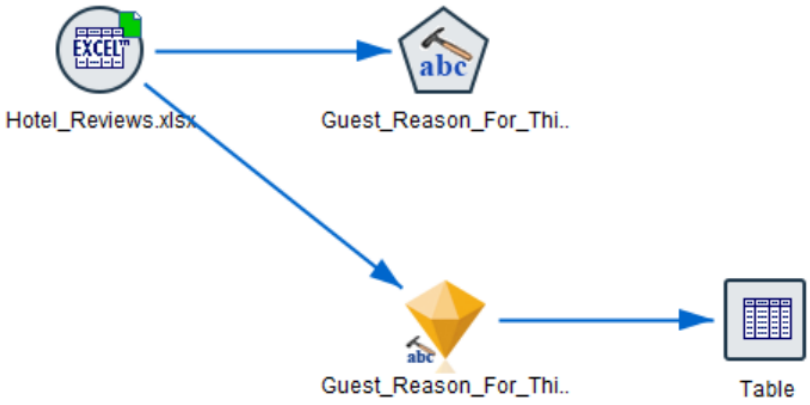


Figure 10.32 The model nugget connected to the hotel reviews data source

Figure 10.33 shows the table node output containing the text model's binary fields that are generated when we execute this stream branch.

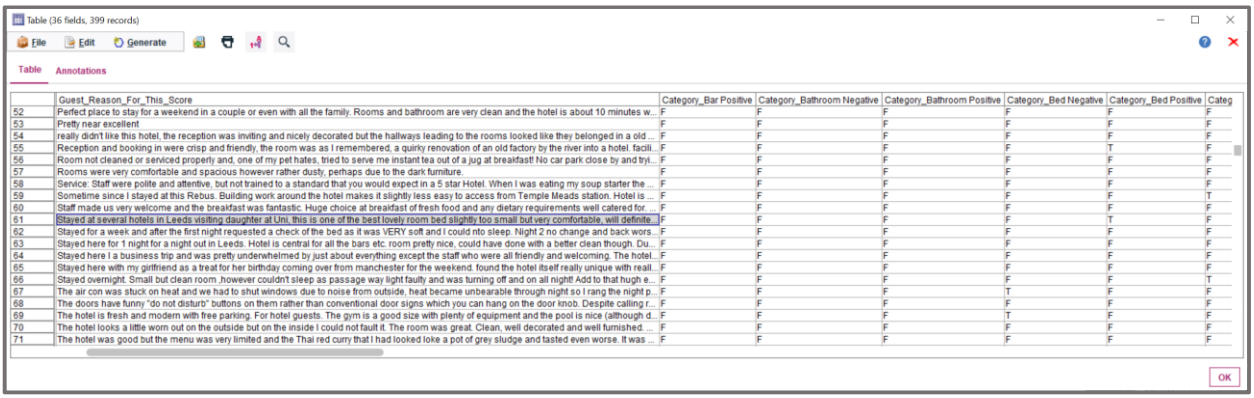


Figure 10.33 Data output from the model nugget showing new binary fields representing the text categories

10.3.1 Testing the model

Having generated a model, we can now test it using some hypothetical responses. The stream file **10_Testing_the_Model.str** contains a data source with a single row response consisting of the following text string:

"The location was great and the reception staff were very welcoming. I thought the room was cramped but the shower was fantastic. They had a cosy bar and did a really good breakfast."

We can pass this example string through the existing hotel reviews text mining model and check which categories the text string contents match with. To illustrate the differences in the outputs generated when using the **categories as fields** mode compared to the **categories as records** mode, the stream contains two separate versions of the model running each mode.

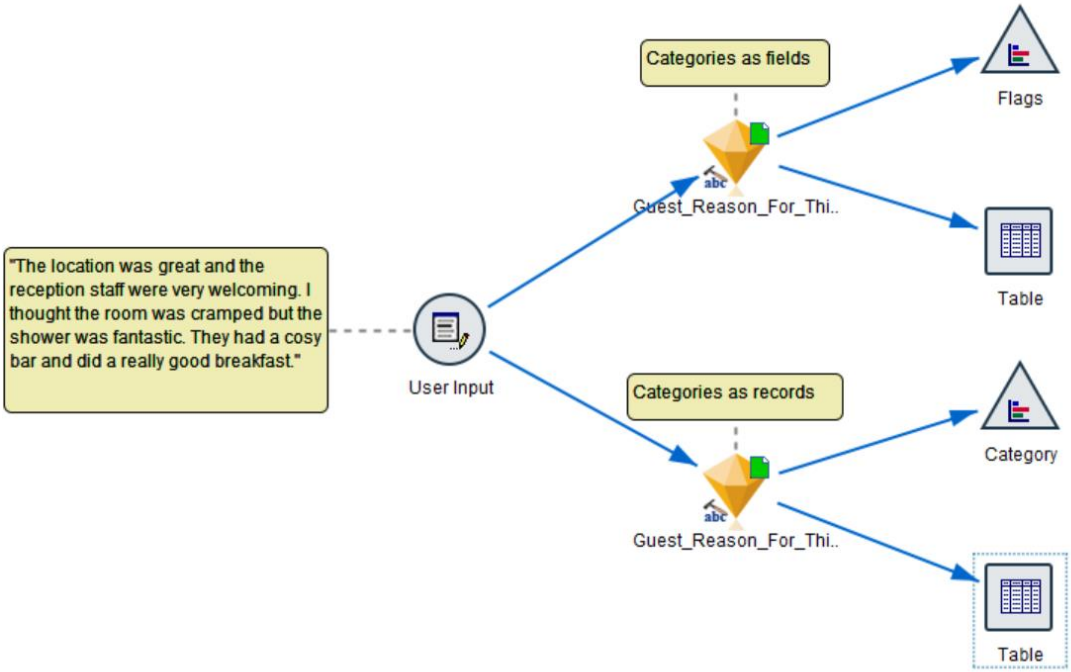


Figure 10.34 Stream containing two versions of the same text mining model to compare categories as fields vs categories as records

Figure 10.35 shows the results of executing the **categories as fields** branch at the top of the stream canvas. As before, this produces a single record with multiple binary fields (referred to as **flag** fields in Modeler). The bar chart shows individual instances of **true** values in these fields. Note however, that in Figure 10.36, which shows results from the categories and records branch, only one additional field (called **Category**) has been created, but 6 separate records are used to represent each matched category in the string statement. This simply serves to illustrate that these two modes generate new data that are represented by very different *structures*. It also demonstrates that the model did a good job of creating categories that matched the main topics and their associated sentiment as true values are returned for **Good Area/Location**, **Reception Positive**, **Room Negative**, **Bathroom Positive**, **Bar Positive**, and **Breakfast Positive**.

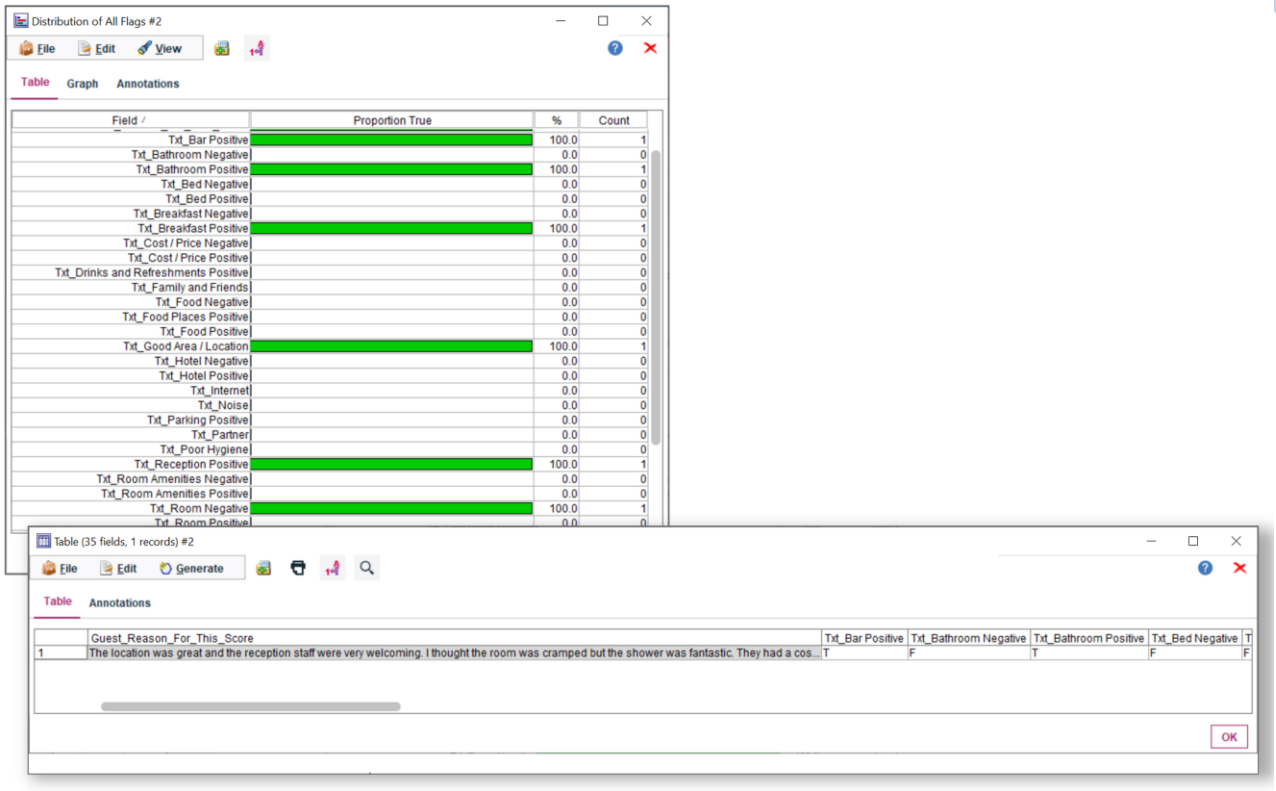


Figure 10.35 Output from the 'Categories as fields' branch of the stream '10_Testing_the_Model.str'

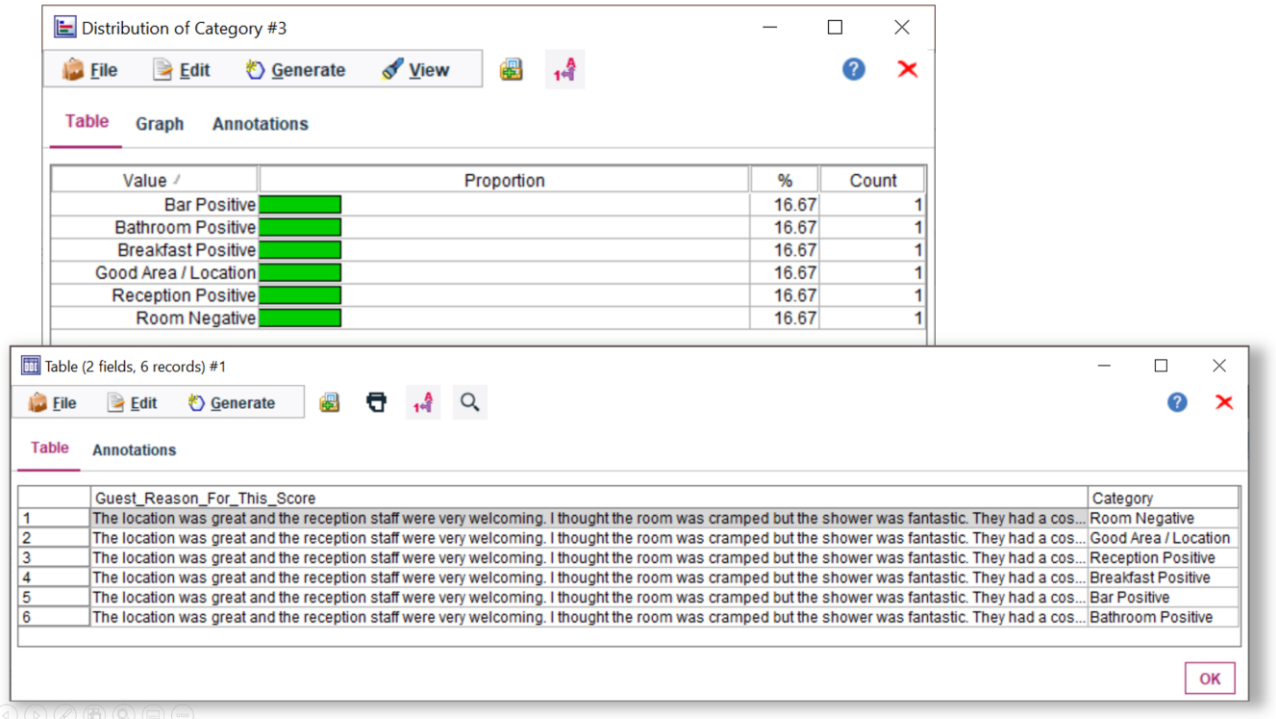


Figure 10.36 Output from the 'Categories as records' branch of the stream '10_Testing_the_Model.str'

10.4 Analysing scored data

Finally, we should remember that the text mining model provides real data that we can analyse in the same way that we might interrogate any existing structured data source. We can see how powerful this approach can be by exploring the contents of the stream file **10_Analysing_Scored_Data.str**. This stream merges a *second* data source that contains a field indicating whether or not the respondents recommended the hotel they reviewed. Figure 10.37 shows that this field splits the 399 records almost evenly between the groups **Yes** and **No/Not Sure** in the recommendation question.

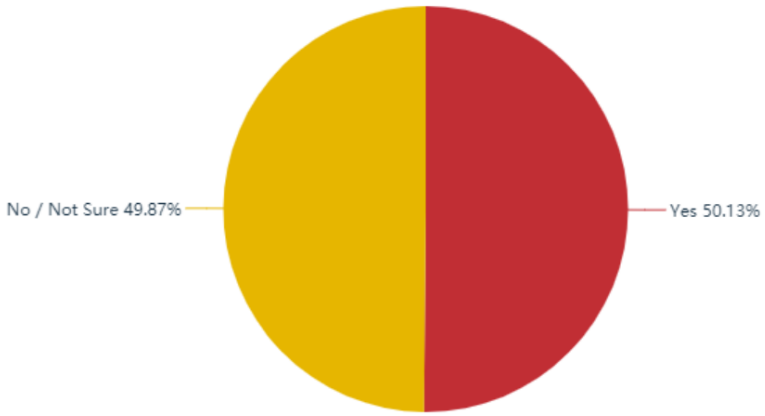


Figure 10.37 Responses to a question asking the guests if they would recommend the hotel they stayed in

Using the **Auto Classifier** node in IBM SPSS Modeler, we are able to build a series of predictive models that use the fields generated by the text mining node to see if it is possible to predict whether the guests would recommend the hotel or not. To assess how reliable the models' accuracy is, we have used a **Partition** node to split the data into **training** and **testing** groups. This means that the models are built on a random selection of records that comprise around 50% of the original data set (so about 200 records). The model accuracy is then *tested* against the remaining 50% of the dataset that was withheld from the model building process. Figure 10.38 shows that the Auto Classifier node tried a range of algorithms with various settings in an attempt to build a model with the highest overall accuracy. The results shown in the image represent the top 5 most accurate models which are ranked in terms of their performance on the withheld test partition. We can see that the overall accuracy ranges from 70% to 66.6% in predicting whether the respondents would recommend the hotel based purely on our 34 text categories.

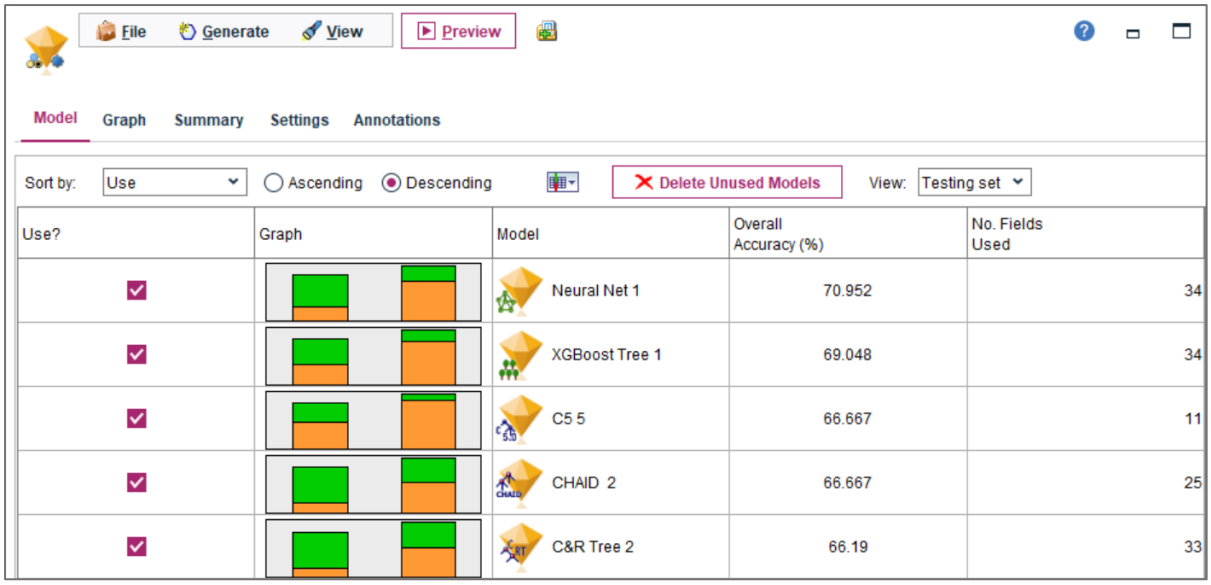


Figure 10.38 Top 5 most accurate models built by the Auto Classifier node at predicting respondent recommendation

Furthermore, when we investigate an individual model, such as the CHAID decision tree, we can view the model's **Predictor Importance** chart. As Figure 10.39 shows, this allows us to see which text category variables are most influential in predicting the outcome accurately. Such an approach can be extremely valuable for individuals who wish to know which aspects of their service provision they need to optimise in order to drive the best outcomes. In the case of the hospitality industry, this sort of model can indicate what hotel managers need to get right in order to maximise customer loyalty.

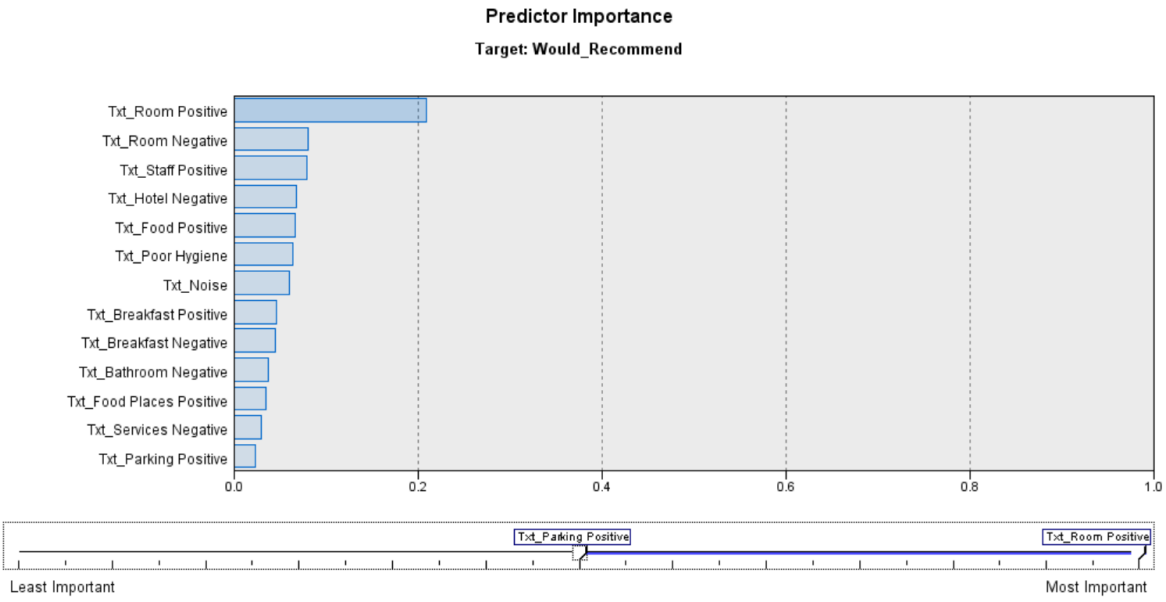


Figure 10.39 Predictor importance chart for CHAID decision tree model showing the most important text categories in terms of driving overall accuracy

Practice Exercise – Chapter 10

Within the folder **Student Exercises** open the following stream:

Chapter_10_Practice.str

1. Right-click on the text mining node and load the text analytics package template **Car_Rental_Modelling.tap** *from the same folder*. Run the node to extract the concepts.

2. Examine the extracted concepts and types. Investigate how well they are categorised by the resources in the TAP file.

3. Create a model nugget from the Interactive Workbench and connect it downstream of the data source node. Edit the model node so that field name extension is now **Txt_** rather than **Category_**.

4. Connect the edited nugget to the downstream **Type** node. Notice that the **Autoclassfier** node is now called **Gender**. This is because the node has been set up to attempt to predict the respondent's gender based on the categories generated by the text mining model nugget. Execute the **Autoclassfier** node.

5. When the **Autoclassfier** has finished, connect the generated predictive model nugget to the final **Matrix** node in the stream and run the **Matrix** node. How accurately has the model nugget predicted respondent gender?