UPPSALA
UNIVERSITET

Locating redundant segments using a translation memory, Transit^{NXT}

Anna Eriksson

Uppsala University
Department of Linguistics and Philology
Språkteknologiprogrammet
(Language Technology Programme)
Bachelor’s thesis in Computational Linguistics, 15 credits

January 10, 2010

Supervisor:
Mats Dahllöf

Abstract

The purpose of this work was to find a method for locating redundant information between chapters of the same manual, by using the translation memory Transit^{NXT} provided by STAR group. A filter for comparing the similarity of sentences, Fuzzy filter, was set to assemble sentences with >50% recemblance and the results were saved. Concordance search—based on the result from the previous step and the terminology lexicon—was set to retreive segments with 80% similarity and to display 300 matches. The results showed that the number of segments with similarities in the reference material that also contained terms were up to 3% (WSM PGRT-TI). The results are lower than STAR translitera's previous method for locating overlapping information, which however was not term dependent. The results open for discussion on how to improve the quality and impact of allowed and forbidden terms respectively.

Contents

Acknowledgement

I would like to thank my supervisor Mats Dahllöf for his guidance, advice and feedback.

I am also grateful to everyone at STAR translitera, in particular Rangnvald Clayhills-Hedemann for his advice, Christer Furberg for support and Tomas Olsson for encourage.

Thanks are also due to Brigitte Orlowski and Karin Dellby for guidance and providing material.

Finally, thanks to Margareta and Tora for proofreading this thesis.

1 Introduction

Technical manuals are typically divided in chapters which contain information on different topics, such as a mounting manual, a maintenance manual and a dismantling manual. Even though companies usually prefer their manuals to be compact, some information tends to occur in multiple chapters, and thus cause redundancy. Locating redundant information is valuable to abbreviate the manuals which can make them more user-friendly and keeps translation costs down. This is the task I have helped STAR translitera to solve, by using their translation memory Transit^{NXT}.

1.1 Purpose

The purpose of my work was to develop a method to locate redundant information, between chapters of the same manual. Never was the search focused on locating redundancy within the same chapter. This limitation is motivated by the difficulty of determining what is redundant within a chapter concerning a specific topic.

The location of the duplicate content was done by using the translation memory (TM) Transit^{NXT} developed by STAR group. The text used in this work is a manual from a Swedish manufacturer of heavy trucks and buses. The company prefers to stay unnamed in this thesis. Nor do they want their manuals content to be displayed here, hence are all the examples fictive.

Redundancy is, in this thesis, defined as segments containing words listed in the company's own terminology lexicon. I was not interested in locating duplicate parts that only contained words with a pure grammatical sense which cannot be relocated or abbreviated. Nor was I interested in finding overlapping but non-redundant parts. Parts that need to occur in the text and are relevant in the context. An example of an overlapping but not redundant phrase, according to my definition, is *Visit a workshop*. The purpose of this definition of redundancy is to maintain all relevant content of the text.

Finding duplicate information with a TM, a task that the system is not designed to perform, required some work to specify the appropriate settings and filters in Transit^{NXT}. The method I developed in Transit^{NXT} to locate duplicate content should not be especially designed for this matter, but general enough to be used to locate duplicate content in manuals from other industries as well.

1.2 STAR translitera

STAR Group provides multilingual, technical communication from 32 countries. The headquarters are in Switzerland and the company is the world's largest privately owned language service provider. The Swedish office, STAR translitera AB, has been located in Uppsala for about 20 years and translates primarily Swedish and English texts, but as a member of the STAR group they can manage translation projects for about 160 languages. STAR translitera in Uppsala are globally responsible for the Swedish customers, and use associated companies as subcontractors for the multilingual translations. In the same way STAR translitera is a subcontractor for the other members of the STAR group, as they need translations to Swedish, Danish or Norwegian. Pivot languages for texts with a Nordic target language are French and German. Besides translating STAR translitera markets and sells the products that STAR group provides in Sweden.

STAR's tools are:

- Transit, a translation memory
- TermStar, a terminology management system
- WebTerm, a system for managing and publishing terminology
- FormatChecker, a quality checker of document formatting
- STAR JamesTM, a system for automated workflow
- ETKE, a networkable database system
- SPIDER, an automated publishing system
- SPIDER Online, a system for publication of product information
- GRIPS, a system for multilingual information management.

The tool I have used in this work is STAR translitera's latest version of the translation memory Transit, which is called Transit$^{\mathrm{NXT}}$.

1.3 Outline

This paper starts with a chapter describing previous work related to this thesis. Chapter 3 describes the resources used in this work, text material and programs. In section 4 the implementation is described in detail, which is followed by chapter 5, Results.
A discussion of the method and results can be found in section 6. Which also contains suggestions on further work.

2 Related work

Manuals are used in many different industries, and it is not surprising that companies want them to be efficient and suited for the purpose but still not too expensive to manage. This is a complex adjustment which can be handled with different approaches. One approach to reduce the costs is to automatize the translation, which is efficient and results in a high text quality with today's technology. By using a TM the translation costs can decrease continuously because the translated material grows and is being reused. Eventually all required material is stored in the memory, and the translation is no longer a factor for expenses.

This chapter presents previous projects on redundancy. Here with the main purposes of detecting redundant documents and analyzing comprehensibility, section 2.1. The following section, 2.1.1, gives a presentation of the benefits of using an already purchased tool for another purpose. The final section gives a brief presentation of the project STAR translitera performed in 2008 with the aim of locating redundant information in a technical manual.

2.1 Previous work on redundancy

Efficiency is essentially about using resources in a way that maximizes the profit. Efficiency in manuals can be seen from at least two perspectives, the writer's and the user's. What is considered an effective manual from the user's perspective is probably comprehensibility. This was at least the thesis Porsche AG had when they turned to Semiotis^3, for evaluating the usability of their manuals, Eybe and Messelken (2009). They constructed a method for analyzing usability by investigating comprehensibility and functionality in two indices, which considered complexity, frequency, length, proportion and structure, and spelling and terminology. One of the goals of this project was to investigate the efficiency of their manuals without investing in a new system. After finishing the project Semiotis^3 recommended a number of actions to improve Porsche's manuals, such as developing a training plan to improve the work of the writers.

Instead of analyzing the efficiency from the writers' point of view, let us investigate the efficiency from the entire company's perspective. This probably has a lot of aspects, but in the end the economic factors are essential to a company. That a manual is easy and cheep to maintain is important, and probably reflects on the usability as well. If two sections of a manual share most of their content it is not efficient to translate each of the documents, since translating is a time-consuming task.

A number of approaches of detecting similar documents have been performed, and there are several algorithms for it. Karl Pearson developed the Pearson product-moment correlation coefficient, partially based on Francis Galton's concept of correlation 1888, Rodgers and Nicewander (1988), to calculate correlation. Cosine similarity measures the similarity by finding the cosine of the angle between two vectors of n dimensions, Salton and McGill (1983). The Jaccard coefficient, by Paul Jaccard, measures similarity by comparing the presence or absence of q characteristics, Allaby and Allaby (1999).

Wan (2008) presents a method for retrieving similar documents by focusing on the structure of a document. The structure is based on the topics and subtopics of a text, which are detected by the algorithm TextTiling. By segmenting with respect to passages based on the topics of a text the semantic passages are located, and the structure of a text is determined. The similarity between documents was detected by analyzing the structure and the number of overlapping topics. Especially documents with parallel main topics that also shared a number of subtopics were calculated as being highly similar. To keep it simple, the more similar passages two texts contain the more similar are the documents.

Another approach to determine whether two documents are similar, or even the identical, is by analyzing the terms in them. By using a phrase recognition program, such as Textract, the terms of a document can be detected. The hypothesis in Cooper, Coden & Brown (2002) is that documents with ''identical lists of discovered terms are identical in content''. Since Textract returns the terms in their canonical form, or root form, terms will be considered identical even if they appear in different forms and versions. In Cooper, Coden & Brown's model, documents can be evaluated according to a number of measures, such as the document name and size, and a document signature based on the located terms. This method could compute the similarity of documents based on terms, and could also recognize documents that contained parts of other documents.

When it comes to using a TM for another purpose than translating I have not found any past research. But creating a method for a TM to perform operations outside of its traditional usage gives an opportunity for the company purchasing the TM to save money. Since a tool that can perform a wide range of operations is likely a positive economic factor. This possibility can also open for a discussion on what other benefits can be retrieved from a string matching system such as a TM.

2.1.1 STAR's previous project on redundancy

STAR translitera has searched for redundant information in manuals before in an internal non published work that I had the priviledge to tale part of. The company TetraPak wanted an approximate measure on how much overlapping information their manuals contained, with the aim of shortening the process of writing new manuals for new products. Christer Furberg and Henrik Hahne, both at STAR translitera, searched for overlapping information using the previous version of the TM, Transit^{XV}. The method was based on the built-in statistics tool in Transit, which returns values such as number of segments that are pre-translated, partially translated, fuzzy and not translated. They calculated weighting based on the company's pricelist and received a value for how much a of manual had to be written, e.g. about 81% for the product manual TT/3 EM_2603215. For example they received a value on how many segments were not overlapping and therefore had to be translated from scratch, and the overlapping value was calculated to 100-81,11 = 18,89%.

In STAR's work they first searched for overlap between different versions of the same manual for the products TT/3 and A3Flex (Table 1). The second search focused on locating overlapping segments in chapters of the same topic between manuals for different products. The results from the second search is shown in Table 2, where the fist column is for product, the second column specify the chapters such as Maintenance Manual (MM) and Operational Manual (OM). They compared the manuals based on the thesis that the larger numbers of manuals in the reference material the more overlapping segments should be found.

In the first comparison, version 1 was not compared to any reference material, v. 2 was compared with v. 1 as a reference, v. 3 had both v.1 and v. 2 as references and so on. They added more reference material for each version, hence the overall increase of overlaps. The second comparison was implemented section-wise, with the first manual of the section without reference, the second manual of the same section used the first one as a reference etc.

Product	Version no	Overlap
TT/3	v.1	23.21%
	v.2	24.41%
	v.3	34.24%
	v.4	24.93%
A3Flex	v.1	20.05%
	v.2	65.91%
	v.3	74.24%

Table 1, Comparison between versions

Product	Section	Overlap
TT/3	EM	18.89%
A3 Flex	EM	3.63%
TCBP20	EM	42.86%
TT/3	IM	1.46%
A3 Flex	IM	11.54%
TCBP20	IM	20.54%
TT/3	MM	
A3 Flex	MM	1.03%
TCBP20	MM	11.42%
TT/3	OM	0.40%
A3 Flex	OM	7.12%
TCBP20	OM	16.32%
TT/3	RM	0.86%
A3 Flex	RM	50.65%
TCBP20	RM	54.05%

Table 2, Comparison between segments for different products

Since they were not sure of how efficient this measure was, they wanted new ideas regarding how to detect the extent of redundant information in a manual, and this is the reason for my method.

3 Resources

This chapter describes all the resources used in this work. The manual I have worked with is described in detail. The size and content of the different chapters are presented and a hypothesis of the result is presented with respect to this. Partial preparation of some files had to be done in order for them to be imported accurately into the TM, for which some freeware was used.

The functionalities of the TM Transit^{NXT} is described in section 3.4.1, along with a picture of the processe and workflow of the system. The terminology handling system TermStar^{NXT} , which I used to handle the terminology lexicon retrieved from the company, is also described in this chapter.

3.1 The Manual

The manual I received from the company to locate redundancy in consisted of four parts, where the workshop manual is divided in three separate parts.

The first section is a driver's manual (DM) with driving-specific information such as driving-unit functions. Naturally this section is intended for the truck and bus drivers. The drivers are probably familiar with the interior of the vehicle such as the control panel but not too familiar with specific parts of the engine.

The workshop manual is the largest section and is divided in three folders: WS, WSM and WSM PGRT, with the purpose of easier handling a large amount of information. WS and WSM both contain information to the workshops about reparations and maintenance. WSM PGRT covers the latest series of trucks, the P- G- R- and T-series. The workshop manual is intended for professional mechanics with much experience and knowledge regarding the various components of a vehicle and their functionality. Since these folders are not separate parts I have chosen not to compare them to each other. The overlap between these folders cannot be seen as an overlap between sections but as internal repetitions which is not taken in to consideration in this work.

The chapter called Technical information (TI) contains information about urgent technical updates or additions to an existing manual. Information that is too urgent to wait for the next version of the workshop manual is updated in TI instead. This means that the content of TI is probably moved to the workshop manual as the new version is written. This further means that the TI is intended for the same readers as the workshop manual, namely mechanics with a in-depth knowledge of vehicles.

The section SDP3 differs from the other sections because the focus of the chapter is on a computer system installed in the vehicles. This section gives guidelines on how to use this system, which makes it possible to program electronics in the vehicle, such as setting a speed limit or detecting malfunctions. I assume that this section is intended for drivers or mechanics, either way the reader is probably not an expert in computational systems and therefore the chapter may be described on a fundamental level.

3.1.1 Size and extent

The sizes of the folders vary largely, from about 9 000 segments in the smallest chapter TI to over 147 000 segments in SDP3. The size of TI is easily explained from its content, urgent updates and smaller additions to the workshop manual, and does naturally not contain huge volumes of information. The largest chapter is SDP3. This chapter's focus is on the computer system that the chapter describes. The extent of this manual can probably be explained by whom the chapter is intended for. If this chapter would have been a part of a manual intended for a computer scientist it probably would not be as detailed and therefore more compact. The workshop manuals differ in size from each other, WS contains almost 130 000 segments while WSM contains 30 000 and WSM PGRT contains 20 000 segments. DM contains about 13 000 segments and is the second smallest after TI.

I received the material in English, German and Swedish. Since I received the complete Swedish manual first I based my work on this language.

3.1.2 Expectations based on the material

When knowing what each chapter contains and how large it is, one can make an assumption on which chapters are likely to contain more redundant segments.

My exptectations are that TI has the largest number of redundant segments when compared to the workshop manuals. TI is, as mentioned, a complement to the workshop manual, which is why it would be natural if they overlap. Because of TI's limited number of segments the redundancy is likely to be larger when WS, WSM or WSM PGRT is used as a reference, and not the other way around.

The assumption about DM is that this chapter shares the most number of segments with the workshop manuals, in the direction with DM as working file and WS, WSM and WSM PGRT as reference material. This could be the case because the content might be quite similar, the workshop manual is more detailed and advanced but the main topic of these manuals are probably the vehicle and its functionalities.

I do not expect to find much redundancy from other chapters in SDP3. This is a special chapter which probably does not deal with the vehicle's physical parts, but the main focus in this chapter is on the computer system.

3.1.3 File format and preparations of the files

The working material consists of six folders, one each for the chapter DM, TI and SDP3 and three folders for the workshop manual.

The section DM was sent to me as Memory Initialization Files (.mif) a format used in project compilation. The .mif files were decoded in the import to the TM Transit$^{\mathrm{NXT}}$. The driver manual was the first section to be used as reference material (see section 3.5.1).

SDP3 was sent to me as .fm files, a format from Adobe Framemaker. This is a file type that Transit$^{\mathrm{NXT}}$ cannot import which is why the .fm files were converted into .mif files before they could be partially decoded by Transit^{NXT}.

.xml was the format I received TI as. These files were decoded by using regular expressions in the program Notepad++, see section 3.2. To be able to import the files in Transit^{NXT} as UTF-8 text files, the files where renamed from .xml to .txt by using the DOS command REN.

The workshop manuals were sent to me as .xml and .fm files, and they were treated as the files above.

3.2 Help programs

To minimize the manual handling , of decoding the files that Transit^{NXT} could not handle and going through the terminology lexicon, some help programs were used.

3.2.1 Notepad++

To be able to easily decode the .xml files in a Microsoft environment, without having much knowledge about the possibilities of decoding via the command prompt in this operating system, I decided to search the web for an appropriate tool.

Notepad++ is a freeware tool which can edit source code and is compatible with the Windows environment. The interface is based on the Microsoft Windows program Notepad, and is therefore user friendly to frequent Windows users. This tool was suited for the task of using regular expressions to locate and remove .xml tags. A task the traditional Notepad cannot handle.

3.2.2 AutoHotKey

I used the script tool AutoHotKeys to parse the terminology lexicon, see section 3.3 and Appendix A.

AutoHotKey is an open-source freeware for Microsoft Windows which makes it possible to automate keystrokes and mouse clicks. With a number of readable commands you can create dialog boxes, manage the volume, monitor your system and automatically expand abbreviations. The script was easy to learn partly because the possibility to follow the execution on the screen.

3.3 Terminology

A terminology lexicon, containing about 600 terms, was also received. I received the lexicon as two different file types, .xml and .xls. I chose to use the .xls file and by exporting the content of this file to plain text it could be imported in the terminology tool.

3.4 The software

3.4.1 Transit^{NXT}

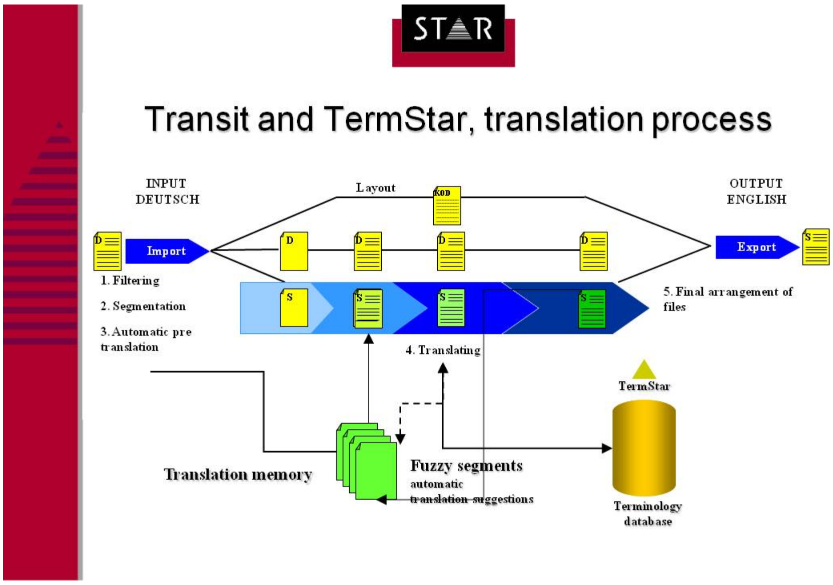
Transit^{NXT} is the latest version of the TM produced by STAR group. The program is a TM which mainly recognizes patterns in the text, but always needs a source text and a target language. The manual I worked with was written in Swedish and the target language was set to English, even though the actual language of both language pairs was Swedish. Because Transit$^{\text{NXT}}$ is a translating tool the program errors if both source and target are set to one and the same language. Setting the target language to English is a workaround that makes the system treat the two texts as written in different languages.

The lexicon—referred to as the reference material—in Transit$^{\text{NXT}}$ differs from many other TMs because the user can decide which files to use as reference material. To translate a text, so the target language would actually be English, Transit$^{\text{NXT}}$ requires reference material in both source and target languages. The system automatically returns pre-translated segments, partially translated segments and fuzzy matches (see section 4.3.1). As all my material was in Swedish—both source and target languages in the working file and the reference material—the segments that were pre-translated were not actually translated into anything and hence not changed.

In this work the sections of the same manual was compared and therefore the reference material was set to one chapter at a time. A detailed schedule of how the comparison was implemented can be seen in section 4.1, table 3.

3.4.2 The features of Transit

Picture A gives a detailed description of the workflow in Transit.



Transit and TermStar, translation process

Picture A, the workflow of Transit

The first steps in the Transit workflow are filtering, segmentation and pre-translation. Transit segments on sentences and after colon and numerated lists. At this step each segment is allotted an identification number. This makes it possible to always see which file the segment belonged to and where in the text it occurs. All translations are made based on the reference material which is selected by the user. This gives the user control of the translation and one can easily look at the reference material to figure out why the system performs certain operations. Fuzzy segments, segments that are partially similar to the reference material, and pre-translated segments can be located. These are constructed by matching segments from the source text to the reference material, an exact match results in a pre-translation. Segments containing terms which aligns to the terminology database TermStar(see section 3.4.3) can be located as well. In the process of actually translating, since this is what the system is designed to do, the user is given all the pre-translated and fuzzy segments and can start the manual work of translating. This consists of reviewing and correcting errors in partially translated segments, and complete translation of non translated segments. Separately from the TM, layout is decoded, which means that the user works basically with plain text.

Transit searches for matching patterns in a database of files, the reference material. This can be seen as a type of lexicon that the user can choose the extent of.

3.4.3 TermStar

TermStar is a terminology tool and works as a complement to Transit. Terminology lexicons can be uploaded in TermStar, the proper lexicon is chosen via Transit to be used on a certain text and the two tools are perfectly compatible. I worked in the latest version of the terminology tool, TermStar^{NXT}. Using this terminology tool one can search for terms in the text, replace forbidden words with allowed ones, access terminology across databases and perform operations on terms in a number of languages.

4 Preparations and procedure

In order to perform my work the material and the programs must be prepared, this is described briefly in section 4.1. along with a schedule for the comparison of the files. Section 4.3 gives an accurate step-by-step instruction of the actual method, FuzzyTerm.

4.1 Preparations in the systems

Since the purpose of my work was not to translate the text all text material was in Swedish. The reference material was set to one section of the manual at a time, and each remaining section was compared against it. The comparison was done according to a schedule, see table 3. Because one redundant segment in chapter x not necessarily answers to one segment in the comparing chapter—it might just as well answer to two or three segments—the comparison between e.g. DM and WS was made twice, with both chapters as reference material.

TermStar$^{\mathrm{NXT}}$ was also prepared. I received the terminology lexicon from the heavy truck company, which was uploaded to TermStar$^{\mathrm{NXT}}$. During the uploading the Swedish characters got corrupted and I looked through the lexicon manually to correct the errors.

Reference material	Comparison material				
DM	WS	WSM	WSM PGRT	TI	SDP3
TI	WS	WSM	WSM PRGT	SDP3	DM
SDP3	WS	WSM	WSM PGRT	DM	TI
WS	DM	TI	SDP3		
WSM	DM	TI	SDP3		
WSM PGRT	DM	TI	SDP3		

Table 3, Schedule of comparison

4.2 Redundancy vs. overlap

Redundancy is commonly described as superfluous repetition, not just the repetition itself, which is supported in the Oxford Advanced learner's Dictionary of current English (1995). This means that overlap or repetition is not synonymous to redundancy, overlap is just the re-occurrence of anything while redundancy is something that is superfluous.

The distinction between these two adjectives is important in this case. A sentence that occurs in two texts and is overlapping is not necessarily redundant. To decide whether an overlapping sentence is redundant or not the content and meaning of it must be taken in to consideration. In this particular text redundant sentences are delimited to the ones containing a term, defined by the terminology lexicon.

4.3 Method - FuzzyTerm

The first step towards receiving the redundant segments is importing the files with the chosen reference material. If the files are not re-imported for every reference material Transit^{NXT} will remember the values from the import with the first reference material, and all operations will be done using that material as a reference. The procedure that follows in Transit^{NXT} can be seen as two separate steps, the Fuzzy match step and the concordance step.

4.3.1 Step 1: Fuzzy match

Transit^{NXT} has a filter called Fuzzy which assembles segments with similar translations. This similarity measurement looks for patterns in the segment as a string, and considers similarities in a number of factors such as words and word order. Depending on the type of differences—transposition, substitution, insertion or removal—the similarity is calculated differently. The fuzzy measurement compares segments in the working file with the selected reference material, hence the segments presented are segments from the working file which have matching parts in the reference file.

In the first step the fuzzy measure was set to >50%. The level of the fuzzy filter was decided upon the facts that a too low value can return irrelevant segments, while a too high values might not capture all the relevant segments. Setting the level to 50% seemed like a good compromize. This step results in that segments from the working folder are compared with segments from the reference folder, and all segments with a similarity rate of 50% or more are assembled and shown on the screen. This result was saved in a single file with the same file extension as the source language files e.g. FuzzyDM-WS.SVE. This relatively low fuzzy level was chosen because I wanted to get a high recall, and not leave out relevant segments. Irrelevant segments will be sorted out in step 2.

4.3.2 Step 2: Concordance search

The purpose of the second step was to further delimit the number of segments to actually relevant redundant parts. Parts that contain content words that carry actual information, in contrast to form words which are purely grammatical, are the parts considered relevant in this work. Concordance search on the terms was used, which is called Dynamic linking in Transit^{NXT}.

This method is absolutely dependent on a rich terminology lexicon. I used a terminology lexicon in TermStar^{NXT} to search for all segments that contained any term from the files that were constructed in the Fuzzy-step. To do this the file from the fuzzy match was opened in the project and used as reference material.

Since the terminology lexicon contained about 600 terms I wrote a simple script in AutoHotKeys (see section 3.3) to speed up the process of searching for each term. The script can be seen in appendix A.

A few settings were specified in Dynamic linking to find relevant segments. The number of shown matching segments was set to 300, to surely retrieve all segments containing a term and not get a false result. Due to a bug in Transit^{NXT} a number of matching segments were not saved properly, no more than 80 matching segments could be saved even when Transit^{NXT} found over 80 segments. In order to still receive a proper result I wrote down the number of matching segments using pen and paper, and added the sum manually. Since the terms of the terminology lexicon were in their primary form, the accuracy of the concordance search was set to 80% similarity, to assure that terms with agreement endings and other morphological differences were captured. The bug of not being able to save more than 80 segments was not related to the concordance similarity measure of 80%.

By using concordance, search segments from the first step that do not contain relevant words are sorted out. This step excludes a number of segments with overlapping content which are actually relevant in the context, i.e. not redundant but only overlapping, such as the phrase *Visit a workshop*, which can be relevant in a number of different surroundings. When this is achieved the text consists of segments which include relevant terms. They also have similar content to other segments from the reference material.

The results from this step where also saved in a new folder, and this was seen as the actual result. The segments from chapter x with similar segments in chapter y was collected, irrelevant segments such as grammatical or non-redundant ones were sorted out and the remaining segments had both similarities to other ones and relevant words.

5 Results

This section contains the result from the redundancy search from the manuals, Fuzzy >50% and, more importantly, FuzzyTerm. A comparison to the results made with STAR translitera's previous method, presented in Related work in chapter 2, is displayed in section 5.2. Finally the efficiency and robustness of Fuzzy Term is discussed.

5.1 Results from FuzzyTerm

The first step towards the FuzzyTerm result was the first step of the method, namely the Fuzzy >50% matching. The result of this step is interesting in the sense that all segments with some overlapping content are presented. The actual redundancy of segments containing terms is not displayed in this table (4), but it is an important step towards the actual FuzzyTerm result.

The first column indicates which chapter has been used as a reference and the second column states the number of segments in that chapter. The number of segments varied from one import to another, I calculated an average value which is the one reported in column two. Columns 3–8 contain the actual value of the redundancy search, where each heading represents the chapter which was compared against the reference material. E.g. Row 1, column 4 declares the number of redundant segments where DM was the reference and WS was the file.

Because WS, WSM and WSM PGRT are viewed as one and the same section—explained in section 3.1— these have not been compared to each other. Nor has the result for each of the manuals been added, since overlapping segments in WS can be the same segments as the overlapping ones in WSM.

Manual (ref)	Segments, tot	DM	WS	WSM	WSM PGRT	TI	SDP3
DM	12 955	-	35.74	69.25	43.66	4.92	0
WS	128 705	3.33	-	-	-	2.16	0
WSM	30 155	11.7	-	-	-	3.68	0
WSM PGRT	19 751	17.21	-	-	-	6.75	0
TI	8 991	31.8	0	64.86	37.67	-	0
SDP3	147 296	2.6	0	0.1	4.02	0.8	-

Table 4, Fuzzy >50% presented in %

The result chart from the method FuzzyTerm declares the number of segments which were similar to segments in the reference material and contained terms, see table 5.

The order of the reference material in this chart does not represent the comparison schedule; this can be seen in section 4.1.

When comparing Fuzzy >50% with FuzzyTerm similarities in the result of WSM with

DM as reference are notable. Since Fuzzy >50% was a first step towards FuzzyTerm the segments in the first step contain the segments from the second step, hence the similarities in column 8, SDP3, with any reference and WS with TI and SDP3 as references.

Manual (ref)	Segments, tot	DM	WS	WSM	WSM PGRT	TI	SDP3
DM	12 955	-	0.67	1.44	0.28	0.14	0
WS	128 705	0.09	-	-	-	0.29	0
WSM	30 155	0.17	-	-	-	0.57	0
WSM PGRT	19 751	0.17	-	-	-	1.28	0
TI	8 991	0.19	0	0.01	3.03	-	0
SDP3	147 296	0.01	0	0	0.12	0.12	-

Table 5, Results FuzzyTerm presented in %

5.2 Comparison to previous method

This evaluation of the manual has been done according to the method STAR translitera used to evaluate overlapping segments in the TetraPak material, described in section 2.2.1. The import log in Transit presents the number of pre-translated, partially translated and fuzzy segments, respectively. These values were printed and calculated according to STAR translitera's pricelist.

The chart below is designed according to the same principle as the charts in the previous section.

The result from Fuzzy >50% and FuzzyTerm differs from the result from STAR translitera's method on some points. But similarities can also be detected, for examplethe lack of redundancy in column 8, SDP3.

Manual (ref)	Segments, tot	DM	WS	WSM	WSM PGRT	TI	SDP3
DM	12 955	-	0.90 %	20.14%	14.50%	3.85%	0.94%
WS	128 705	21.73%	-	-	-	15.07%	0.94%
WSM	30 155	18.81%	-	-	-	2.66%	0.94%
WSM PGRT	19 751	18.38%	-	-	-	8.34%	0.94%
TI	8 991	15.75%	0.90%	14.39%	19.29%	-	0.94%
SDP3	147 296	19.11%	0.90%	19.71%	20.49%	6.66%	-

Table 6, Results according to the STAR translitera method, presented in %

5.3 Evaluation

Because of the number of terms in the lexicon and the simplicity of the AutoHotKey script, the method was very time-consuming.

To evaluate FuzzyTerm the method was performed backwards, to see if the same result was concluded. The backward FuzzyTerm test was performed on TI, with WS as the reference material. Initially both chapters were searched through for terms, using the same method and settings in Dynamic linking as in the original method. The results for these searches were saved in the folders TermsInTI and TermsInWS.

TermsInWS was specified as the reference material and TermsInTI was opened in Transit^{NXT} and finally the fuzzy filter was set to >50%. The result from this experiment was 0.29% matches, i.e the same result as in the original FuzzyTerm method.

A few random samples were collected and reviewed to estimate the precision of FuzzyTerm. They were collected by picking some folder from the results folder and selecting the first 8 files, each separate from each other. In the review presence of terminology were considered. The samples were collected from the the results of the comparision between DM and SDP3, with the later as a reference material. 47% of the retreived reduntant segments were reviewd, of which 100% did contain terminology. This gives futher weight to the assumption that the method did not result in a large number of irrelevant matches.

6 Discussion

Duplicate information can probably be found in a manual using a number of different procedures, even when restricting the method to using a TM. This can even be seen in this paper, with the method I constructed and the method STAR translitera had already constructed. The advantages and disadvantages of the method FuzzyTerm is discussed in section 6.1. The result is explained in section 6.2.

This work has merely been started and there are a number of directions to choose for further work. A few possible continuations on my work are presented in section 6.3.

6.1 The method

The method is constructed to retrieve as many accurate matches as possible, to give a high precision. I attributed great importance to the terms, which were considered to be the guarantee against claiming parts that need to occur in the text and are relevant in the context to be redundant, see section 1.1. The terminology lexicon was a good source of relevant content words, in contrast to simply removing segments with functions words. This task could have been performed by constructing a list of function words that were not allowed in the sentences. That method would though focus less on the existing TM than the FuzzyTerm procedure does.

A classical problem in computational linguistics, which also occurred in this project, is ambiguity. Some terms could have a second meaning that is not a term, this was the case of the word *ground* which can be a noun or a verb. The first of these two is probably not considered a term in the heavy truck and bus business. But if a segment passed through the fuzzy filter in the first step, it will pass through the second step if it contains a word which can be seen as a term, ambiguous or not. Another problem when it comes to handling segments with terms is duplicate occurrences of terms in the same segment. Consider the segment ''Oil and hydraulic fluid should be refilled''. Oil is a term and hydraulic fluid is a term; when searching through a number of segments with Dynamic linking this segment will be found twice and therefore adding to a false result. This could probably be prevented in Transit^{NXT} by looking at the source file and the identification number of the segments, described in section 3.4.2, and deleting duplicates. This must be performed carefully though; there is a risk of deleting non-duplicate matches and it is possible that the risk is greater than the number of misleading duplicate occurrences.

The files that were sent to me and imported in Transit^{NXT} as .mif files were not properly decoded by the program. Some tags were still in the texts and might have had an impact on the result. Luckily the tags were not too many and occurred consistently in the same type of segments. If this had an impact on the result, the number of segments that were affected is limited.

The method was very time consuming. But with a more robust and powerful script to work through the terminology lexicon faster this method could well be used to locate redundant sentences which contain terms.

The experiment that measured the robustness, in section 5.3 in the Results chapter, showed that exactly the same result was concluded even when performing the method backwards.

The possibility of re-performing tasks, without having to redo the entire operation, is exceptional. As a result of saving files as the work continues, the risk of losing important information decreases. And when an error occurs there is no need to re-perform the entire procedure, which makes the method a little less time consuming.

6.1.1 Problems in Transit^{NXT}

Tranist^{NXT} is a translation memory and is not designed for the task of locating redundancy. But because of its qualities as a string matcher similar strings can be detected.

Most of the problems existed because of the attempts to create something the system was not intended for. The solutions to these problems were often created in contrast to how a program should work. Another general problem was discovering what was possible to do in the system when the purpose was not to translate.

Transit^{NXT} is the latest version of the TM from STAR translitera and does still, at the time of this thesis, have a few bugs.

Two problems in Transit^{NXT} that might have had an impact on the result were detected.
1. The segmentation, explained in section 3.4.2, was different from one import to another of the same material. This was discovered when looking at the total number of segments from the import log, where the number of imported segments differed no more than 1 percentage point. The problem was worked around by calculating the average value of the number of segments, and presenting this number as total number of segments in the results.
2. The import of SDP3 raised a few questions as well. When analyzing the import log for SDP3, with any reference material, a number of fuzzy matches were identified. But when using the fuzzy filter on the entire file, as a first step in FuzzyTerm, no fuzzy matches were found. This is a dilemma with no obvious explanation. One possibility is that Transit^{NXT} performed the operation incorrectly.

6.2 The results

The expectations were based on the content of each chapter of the manual in section 3.1.2. Namely; TI with the workshop manual as the reference material would have the largest number of redundant segments. The largest number of redundant segments from DM was also thought to occur when the workshop manual was used as a reference. Finally SDP3 was not expected to have large amounts of redundant segments with any manual used as reference.

These expectations were proven to be partially true. SDP3 had no redundant segments containing terms when using FuzzyTerm. Both DM and WSM PGRT had the largest amount of redundant segments when TI was used as a reference. TI had

the largest number of redundant segments overall even when compared to SDP3, the exception was when DM was used as a reference. Worth noting is that WSM PGRT and TI had the most located redundant segments, in percentage, when compared to each other.
When WS is compared to TI as a reference the number of redundant segments is 0. This can be explained by the number of segments in WS, 128 705 segments, in comparison to 8 991 segments in TI.

It is possible that a comparison using the English manuals would have resulted in slightly different numbers. Some manuals contained English sentences, which made segments consist completely of English which naturally did not have fuzzy matches with the mainly Swedish texts.

I claim my result to have a high precision, because of the strict screening in the term concordance step. As mentioned in the previous section some errors might have slipped through, but the main improvements of the result can probably be made in recall. In the case of attempting to correct the errors, time efficiency is important, in this already time consuming procedure. The number of false negatives—non located redundant segments—are reasonably the minor part of the segments not considered redundant from the FuzzyMatch. Hence it would be time effective to create a new method for verifying the true negatives instead, on the contrary to FuzzyMatch.

6.2.1 Comparison of results

Locating redundancy in this manual can be done in different ways, and depending on the method different results can be reached, see the chapter Results. This does not mean one of the results is a miscalculation, it means that different approaches have been used and factors have been of different weight.

When importing a text to Transit^{NXT}, and a reference material is used, some segments will likely be pre-translated. These segments are logged in the import log as pre-translated but they are also considered 100% fuzzy matches and are a part of the result in Fuzzy >50%.

A small part of the segments in Fuzzy >50% contains terms, proven in FuzzyTerm. There is also an imbalance between the number of segments with a fuzzy match and the number of segments with both a fuzzy match and a term. This can be explained by the content of a manual, since they handle topics that are more or less term rich. A possible risk is that forbidden terms are used. The company whose manual I have analyzed does have a language review tool, which is programmed to use allowed terms instead of a number of pre-defined forbidden ones. One of the terminologists of the company claims that the language review tool probably is not used to a large extent, Informer 1. An attempt to evaluate the usage of forbidden terms or the usage of the language review tool might be a second step for the heavy truck company.

Another possible explanation for the lack of terms in the manual is that the texts are not compact enough. If this is the case the manuals can be much abbreviated, with the effect that the translation costs can be lowered. The share of segments from TI with WSM PGRT as a reference with terms is 19% (253 segments containing terms of 1 333 segments located with a fuzzy match). This means that the number of segments containing terms throughout the entire manual can be estimated to 3 753 segments.

6.3 Further work

Two main approaches can be used when continuing my work on this topic; optimizing the method or investigating the usage of forbidden terms.

When optimizing the method the main goal should probably be to increase the recall. This could be done by enlarging the terminology lexicon or reviewing the fuzzy matches.

Since the difference between the results of Fuzzy >50% and FuzzyTerms where surprisingly large and non-consistent between chapters it would be interesting to evaluate the usage of forbidden terms.

In WSM when TI is used as a reference material 5 832 segments where found as fuzzy matches with Fuzzy >50%. When the restriction of just considering segments with terms as redundant this value was dramatically lowered to 1 segment. This could indicate that forbidden terms are used. An investigation of this could initially consist of searching through the results from Fuzzy >50% for forbidden terms. The heavy truck company does have a list of pre-defined forbidden terms to search through the material for, but there might exist synonyms of the terms that are not defined yet and more forbidden terms needs to be defined. By searching for synonyms to terms on the Internet, in subject-specific lexicons and asking experts one might detect new not yet defined forbidden terms. These words can then be used for concordance search in Transit^{NXT}, and easily be replaced with the proper allowed term. Of course the new forbidden term needs to be added in the company's language review tool as well.

Another explanation to the lack of terms in the text is that the number of defined terms is not high enough. There might be many words in the manual that are specific for this subject with the quality of a term but it is just not defined as one yet. By defining more terms' synonyms to these could be forbidden and a more homogeneous language could be created, the translation costs could be reduced and references to the same component in another chapter would be simplified. Also FuzzyTerm would give a more interesting result.

7 Summary

The purpose of this work was to develop a method using the TM Transit^{NXT} to locate redundancy between chapters of the same manual for a heavy truck. The redundancy was restricted to sentences that were >50% equal to their corresponding sentence and contained a term. Since this should be done using a TM, which had not been tested outside of STAR translitera, the method was not trivial to accomplish.

By using a fuzzy filter—an algorithm that considers insertion, removal, substitution and transposition of words in sentences— set to >50% similarity, the segments with similarities in the reference material were assembled. In this step each chapter of the manual was compared to each other, one by one. The result from this step, Results section 5.1, corresponds partially with the method STAR translitera used in 2008 to retrieve overlapping information in manuals from TetraPak.

The second step in the method was assembling of the segments containing terms. This was done with concordance search using the terminology lexicon, based completely on the files created in the previous step. Each occurrence of a term was counted and saved in a unique file. When both steps were completed the method was implemented and the result could be calculated.

The results showed that the number of segments with similarities to segments in the reference material and containing terms were up 3% (WSM PGRT-TI). The effectiveness of the procedure is low. An experiment showed that FuzzyTerm does generate the same value when performed in reverse. The precision of the result is estimated to be quite high—from performing random samples of the output—while the recall is estimated to be lower than desirable. The results might indicate that terms are not used as frequently as expected and a number of solutions to that problem is presented in the section 6.3.

A AutoHotKeys script

An explanation of the script `Termscript.ahk`

Activates the window TransitNXT

```
WinActivate Transit NXT
```

Highlight the term on top of the page

```
click 214, 197
```

Loop 100 times

```
Loop 100
{
```

Right click on the highlighted term

```
    click Right
    sleep 1000
```

Press down arrow 14 times, this highlights the dynamic linking link

```
    Loop 14
    {
    send {down}
    }
```

Select the term

```
    send {right}{enter}
    sleep 1000}
```

Move mouse to the save button

```
    MoveMouse 472, 116}
```

Creates a message box with the question "Do you want to continue?". If the answer is no the script close the dynamic linking window and goes to the next term.

```
    MsgBox, 4, Do you want to continue?
    IfMsgBox No
    {
    WinClose Dynamic Linking
    }
```

If the answer is yes, the term is saved in a folder specified for the project. And the dynamic linking window is closed.

```
    else
    {
    click 472, 116
    sleep 2000
    send
N:\projects\Scania\scania_ws_txt\concordance\
    KeyWait enter}
    WinClose Dynamic Linking
    }
    sleep 3000
```

The down arrow is pressed which makes the system move on to the next term, and the loop starts again.

```
    send {down}
    click
    sleep 1000
}
```

Bibliography

Eybe, Angelica & Messelken, David (2009). Comprehensibility as an economic factor. *TCWorld*, April 2009, pp. 22–24.

Rodgers, Joseph Lee and Nicewander, W. Alan (1988). "Thirteen ways to look at the correlation coefficient". The American Statistician 42: 59-66.

Salton, Gerard., & McGill, Michael J. (1983). Introduction to modern information retrieval. Auckland, New Zealand: McGraw-Hill.

Allaby, Ailsa and Allaby, Michael (1999). "Jaccard's index." A Dictionary of Earth Sciences. Encyclopedia.com (electronical).

Wan, Xiaojun (2008). Beyond topical similarity: a structural similarity measure for retrieving highly similar documents. *Knowledge and information systems*, vol. 15: 1 , pp. 55–73

Cooper, James W., Coden, Annie & Brown, Eric W. (2002). A Novel Method for Detecting Similar Documents. *Proc. of the 11th Inter. Conf. on Information and Knowledge Management*, pp. 245–251.

Hornby, A S (1995). Redundancy. *Oxford Advanced learner's Dictionary of current English*, fifth edition, pp 978.

Notepad++ (electronical). `http://notepad-plus.sourceforge.net/`

AutoHotKeys (electronical). `http://www.autohotkey.com/`

Memory Initialization File (.mif) Definition (electronical). `http://www.altera.com/support/software/nativelink/quartus2/glossary/def_mif.html`

Informant 1: Employee at the heavy truck company.