University of Twente
Faculty: Electrical Engineering, Mathematics & Computer Science
Department: Computer Science
Group: Language, Knowledge & Interaction

Natural answer presentation through revision of syntactic patterns

Author: E.J.W. Dikkers

Date: November, 2004

Examination Committee:
dr. M. Theune (1st supervisor)
prof.dr.ir. A. Nijholt (professor)
dr.ir. H.J.A. op den Akker
dr. D.K.J. Heylen

Preface

As final step within my Computer Science degree, I got the opportunity to complete my master within the Language, Knowledge & Interaction (TKI) group in Enschede, The Netherlands. Writing my master thesis covered the period from the 2^{nd} of February 2004 till the 26^{th} of November 2004 and this thesis describes my work and experiences during this period. Within my Computer Science degree I specialized myself towards Language, Knowledge & Interaction because I really find it fascinating to be able to make the communication between man and machine more simple by employing techniques from Artificial Intelligence and Language Technology. We hear regular human language everywhere around us when people are communicating with each other, so it would be great to combine the techniques from these two fields for "natural" communication.

In Enschede, within the TKI-group of the University of Twente, I explored a sub-field of the state-of-the-art research field of "Question & Answering" focused at the natural presentation of answer texts. Once in a while we all have questions that we want to be answered, but due to information overflow it sometimes seems impossible to find a satisfying answer within a reasonable amount of time. This thesis is especially oriented at the way that retrieved information is presented as an answer in the end. During this research I'll try to come up with an algorithm for presenting answer texts. I'll try to realize this by a thorough literature survey regarding the matching of questions & answers, accordingly I'll test by means of an experiment, this will result in a design & implementation phase in which I'll try to come up with an algorithm to show the practical usage of the underlying pattern and I'll finish with an end evaluation of the implemented program. All in all my general goal is to walk through the different research steps, I have learned during the theoretical phase of my Computer Science degree, in order to achieve a satisfying result.

I'd further like to thank all the people that made it possible for me to finish my Master in Computer Science. I'd like to thank my examination committee (Mariët Theune, Rieks op den Akker, Anton Nijholt & Dirk Heylen) for steering and providing me the necessary feedback to realize this master thesis. Further I'd like to thank the University of Groningen for providing me the necessary Alpino annotated materials used within my research. Also I'd like to thank all the students I have met during my degree, because you all made it a time to never forget. I'd like to thank my family, friends and parents, who made it possible for me to finish my degree in my own tempo. And last, but by no means least, I'd like to thank my beautiful girlfriend Annette Beukers, for providing me a save heaven to return to in times I was not so confident in realizing this thesis. Thank you all for your support and confidence in me!

Enschede, The Netherlands, 24 November 2004.

Erik Dikkers

Abstract

This thesis "**Natural answer presentation through revision of syntactic patterns**" is focussed at answer presentation as sub-field of the broader "Question & Answer" field. The main research question is how properties of a question, properties of a retrieved text fragment and relations between questions & answers can be used to present sentences that are as natural as possible and forms an answer for a question. All this because a retrieved text fragment often needs revision, to be interpreted as a natural answer. By revising a "raw" retrieved text fragment, as delivered by an information retrieval agent, it becomes a natural answer for the question. A first attempt for the revision of text fragments is based on the properties of questions, the properties of fragments and the identification of the 'topic-comment' pattern as pattern for an answer to generate. The properties and the pattern, are based on theoretical research and the hypotheses, related to the topic-comment pattern, have been confirmed in an experiment. To show that it is possible to generate answers, based on this pattern, a demonstrator has been implemented. The demonstrator shows the found properties and shows that it is possible to generate answers based on the topic-comment pattern. But it also shows that the generation algorithm is not working for every question and answer pair. What can be concluded is, that answers based on the topic-comment pattern are accepted as natural answers. Besides that, there can also be stated that work oriented at better generation of topic-comment based answers and work oriented at other means of text revision is necessary. All this to present a natural answer in a question-answering system.

Samenvatting

Dit verslag met de titel “Natuurlijke antwoord presentatie door het herzien van syntactische patronen”, gaat in op antwoord presentatie als onderdeel van het grotere “Vraag & Antwoord” onderzoeksgebied. De belangrijkste onderzoeksvraag hierbij, is hoe eigenschappen van een vraag, eigenschappen van een gevonden tekstfragment en relaties tussen vragen & antwoorden kunnen worden gebruikt voor het presenteren van een zo natuurlijk mogelijke zin als antwoord op de vraag. Dit, omdat een gevonden tekstfragment vaak herziening nodig heeft, wil het als natuurlijk antwoord geïnterpreteerd kunnen worden. Door het herzien van een ‘ruw’ tekstfragment, kan deze worden omgezet in een natuurlijk antwoord op de vraag. Een eerste poging voor het herzien van tekstfragmenten is gebaseerd op de eigenschappen van vragen, de eigenschappen van fragmenten en de identificatie van het topic-comment patroon als patroon voor een te genereren antwoordzin. De eigenschappen en het patroon, zijn gebaseerd op theoretisch onderzoek en tevens zijn de hypotheses, gerelateerd aan het topic-comment patroon, bevestigd in een experiment. Om aan te tonen dat het in de praktijk mogelijk is om antwoorden te genereren op basis van dit patroon is demonstratie progamma geïmplementeerd. Dit programma toont aan dat het inderdaad mogelijk is om antwoorden te genereren, gebaseerde op het topic-comment patroon. Wat zeker geconcludeerd kan worden, is dat antwoorden gebaseerd op het topic-comment patroon als natuurlijk worden ervaren. Echter, genoeg toekomstig werk, gericht op betere generatie van topic-comment antwoorden, maar ook gericht op andere vormen van tekst revisie is nodig. Dit werk is nodig om uiteindelijk een goed antwoord te kunnen presenteren binnen een vraag-antwoord systeem.

Contents

Chapter 1

Introduction

1.1 Relevance

As more and more information becomes available in our modern society, locating useful information can be very time-consuming. A good solution for this information overflow would be a way of retrieving requested information automatically. The issue for automatically finding relevant information in these large amounts of data is addressed by the research area of information retrieval (IR). One particular branch within the IR field is question answering (QA) which enables users to submit full natural language questions, as opposed to the keyword-based queries that are commonly used in document retrieval and provides answers to users instead of whole documents. The main advantage is, that people immediately get presented the answer they need and don't have to extract it out of the retrieved documents themselves.

To date, most of the work in QA has been involved in *answer extraction* (locating an exact answer in a text collection) and less in the presentation of that extracted text. However for better and natural human-computer interaction it is useful to look at *answer presentation* (a means of presenting the raw retrieved material). It can certainly improve the interaction between QA systems and end-users as QA systems will tackle more difficult questions and will (eventually) be extended to dialog processing systems. See for example [Dikkers, 2003, Kosseim et al., 2003] for papers that point at this issue. With this vision in mind a raw text snippet or a very short answer will not be enough to communicate naturally with a user. A full answer sentence or larger text fragment, that is linguistically motivated, matches features of the question and that is extended with multimodal means of presentation, will be required.

To investigate this area, the Interactive Multimodal Output Generation (IMOGEN) project started within the Parlevink group at the University of Twente in December 2003. The project is funded by the NWO (Netherlands Organization for Scientific Research) in their research program for Interactive Multimodal Information Extraction (IMIX). Its goal is to develop a multimodal information-presentation module combining Dutch language, speech and graphics. The IMIX project covers all the modules necessary to create a working QA-system and the modules responsible for the extraction of answers are particularly interesting. The modules responsible for answer extraction, ROLAQUAD $->$ University of Tilburg and QA-DP $->$ University of Groningen, provide the text fragments that the IMOGEN project will revise. The final result of the IMOGEN project will be a demonstrator that consists of a number of interacting sub modules, dealing with text revision (adaption of text to become in agreement with the previous discourse), multimodal generation and speech synthesis.

This research focuses on the revision of text part of multimodal interaction with a further specialization towards natural answering of a user's question in a question answering system. The answer extraction modules will be developed by the ROLAQUAD and QA-DP project within IMIX and

deliver "raw" text fragments that contain an answer regarding questions asked within a medical domain. The text revision will be based on the adaption of syntactic structures that can be identified through the Alpino-parser and based on syntactic aggregation[1]. "Alpino is a wide-coverage computational analyzer of Dutch which aims at accurate, full, parsing of unrestricted text. It produces dependency structures, providing a reasonably abstract and theory-neutral level of linguistic representation" [Bouma et al., 2001]. It is a parser implemented at the University of Groningen and used for their module in extracting answers for a QA-system. By looking into means of text revision for answer presentation, a new field receives attention. Mainly due to the fact that most work within QA has not been performed for the generation of "natural" answer-sentences, but has been performed on exactly locating these answers through answer extraction.

1.2 Research

The goals of this research are best described by presenting the main research questions, and by presenting an overview of the necessary objectives to be able to provide an answer for the research questions.

1.2.1 Research Questions

The main issue is the question **how the syntactic patterns in a text fragment can be adapted to produce a natural presentation of answer texts?** In order to be able to provide an answer for this question, the following sub-questions have to be answered:

- Which features are important related to "simple" questions and corresponding answers in order to generate congruent question-answer pairs?
- How can text fragments be revised without losing a correct structure?
- Where can aggregation be applied using syntactic patterns?

Since this research is part of a multi-modal research project (IMOGEN) it would also have been interesting to look at constraints of multi-modality that affect the generation of a natural text response. However, since this forms not the main issue to be explored in this research and since there is always the pressure of time, no further time has been invested in identifying these constraints.

Definitions

To provide a better context in which these questions can be placed it is necessary to specify a few definitions. Certainly, since the term "natural" can be interpreted in a lot of ways. The concepts and definitions used are the following:

- **Syntactic pattern**
  This is the dependency structure generated by the Alpino-parser. Both the input question and the input text fragment have to be annotated with at least these dependency structures (See chapter 5 for more information regarding dependency trees).
- **Natural presentation**
  This is the answer provided to the user after revision of the text fragment and after applying aggregation. It consists of a sentence that provides the answer for the original question and has the "topic-comment" pattern as underlying foundation.

[1]The process in which two or more linguistic structures are merged together to formulate a new structure [Shaw, 2002, Reape and Mellish, 1999].

- **Text fragment**
  Part of text that is retrieved material (text snippets, sentences or paragraphs that are the result of answer extraction) and contains the information asked about in the question. The fragment is "raw", no pattern revision revision or aggregation has been applied yet.
- **Topic-comment pattern**
  The order in which the already known information from the question is repeated before the new contributing information out of the text fragment.
- **Congruent pair**
  A question & answer pair that is in harmony conforming the requirements between questions and answers.
- **Correct structure**
  A syntactic structure that does not get corrupted through revision and matches the Dutch grammar regarding sentence structures. This points mainly at the topic-comment pattern, but also indicates the correct usage of plural and singular verbs.
- **Aggregation**
  The process in which two or more linguistic structures are merged together to formulate a new structure.
- **Simple question**
  A question that is restricted in context and can be answered by a one sentence long answer that is accurate and complete. (See for some examples table 1.1)

English question	*Translated to Dutch*
"Can RSI be cured?"	"Is RSI te genezen?"
"What is the definition of RSI?"	"Wat is de definitie van RSI?"
"Which factors are responsible for RSI?"	"Welke factoren zijn verantwoordelijk voor RSI?"
"What are symptoms of RSI?"	"Wat zijn symptomen van RSI?"
"etc . . . "	"etc . . . "

Table 1.1: Simple question examples

1.2.2 Research Objectives

The goals of the project are to explore the question & answer features that are important for realizing congruent question & answer pairs, to explore the use of aggregation within sentences and to explore the use of dependency trees for text revision. To reach these goals we need to do the following:

Text revision By parsing the question and the text fragment with the Alpino-parser [Bouma et al., 2001], dependency structures are available. It is the assumption, that it is possible to realize congruent question-answer combinations through revision of these dependency structures [Anaya and Kosseim, 2003]. Which features play a role in the necessary transformations and how the use of aggregation [Shaw, 2002] has its effect on transformations, has to be explored in order to realize a working algorithm. By completing this objective, it becomes clear that the example in table 1.2, on the next page, can be called a "congruent" question & answer pair.

Example: "Wat zijn [symptomen van RSI]$_{pattern}$?"
Source: "*< answer > De symptomen van RSI zijn uiteenlopend van aard. Ze kunnen variëren van, koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid. < /answer > Ze kunnen zich voordoen in nek, schouders, armen, polsen, handen en in heel uitzonderlijke gevallen de benen. Ook kan het lastig zijn om de pijn te lokaliseren.*"
Output: *"[Symptomen van RSI]$_{pattern}$ zijn uiteenlopend van aard [en]$_{aggregation}$ kunnen variëren van, koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid."*

Table 1.2: Example question-answer pair

Corpus creation To complete the above objective, it is necessary to build a corpus form collected material of internet. Corpus building is necessary, because the modules within IMIX responsible for answer extraction(ROLAQUAD, QA-DP) have no working modules at the time of this writing. This corpus has to contain a set of questions and accompanying text fragments with answers regarding the RSI domain. To identify surface patterns, question and fragment properties, and to use this corpus in the end for computational use, all the corpus material has to be annotated by Alpino. Since Alpino's output is in XML, it is a logical step to realize this corpus in XML. A nice side effect is that the corpus is suitable for automatic processing in this format. During this collection phase of material, constraints related to the quantity of questions and text fragments, the type of fragments (text snippets, sentences and paragraphs) have to be taken into account. This to be sure to build a corpus that represents a broad range of "simple" questions and also contains text fragments with different underlying patterns.

1.3 Research Approach

The approach is based on information out of [Verschuren and Doorewaard, 2000], which provides a clear overview of how to setup a research. This approach can best be described by the picture shown in figure 1.1 and by a short description of each phase (represented by a rectangle) within the picture.

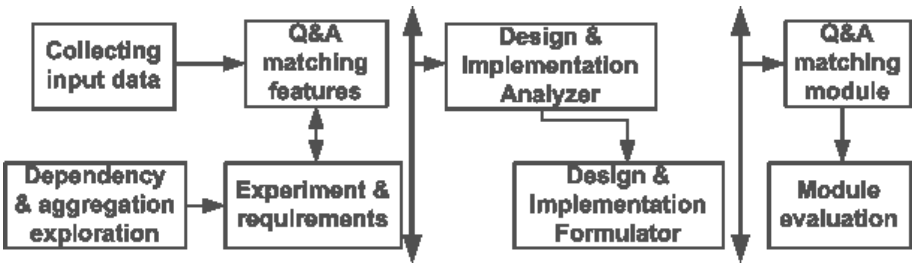

Figure 1.1: Approach overview

During the *Collecting input data* phase, a way of using input data has to be explored to come up with a suitable format for the corpus. This corpus will consists of questions and text fragments from the internet regarding the RSI domain. How the corpus was exactly build, will be explained in section 2.2.3. All this material was annotated by the Alpino parser and that resulted in a corpus consisting of XML documents. Each XML document contains a one sentence long string with its dependency structure and represents a question or a retrieved text fragment. The corpus is initially

necessary for the extraction of patterns, but is also suitable for use within the implemented "proof-of-concept" demonstrator. The demonstrator uses the questions and fragments in the corpus to demonstrate the generation of answers based on the topic-comment pattern.

The next phase *Q&A matching features* indicates the necessary investigation of properties & features of questions & corresponding answers. These pairs are restricted to certain features, patterns and types of answers for example, and need to be identified to generate congruent question & answer pairs.

More or less in parallel, *Dependency & Aggregation exploration* focuses on how to use the dependency trees that are available as annotated material. This phase also looks specifically into possible ways of applying syntactical aggregation. A thorough literature survey is necessary to tackle these two issues.

Based on these surveys and the result of the *Q&A matching features* phase, it is possible to test all the found information through an experiment. The outcome of this experiment is important, because it confirms or disconfirms the hypotheses that will be used for the *requirements* phase: the phase that represents the design and implementation phase of the demonstrator.

The *Analyzer* phase is the phase in which the module, capable of retrieving the important properties concerning questions and fragments, will be implemented. The analyzer has to identify the elements that are necessary for answer generation (elements and attributes within the dependency trees, type of text fragment, etc). More of these properties are identified and further specified during the *Q&A matching features* phase.

The *Formulator* phase is responsible for the Formulator module that generates the answer, based on the properties & features provided by the Analyzer. The answers generated at this final stage are the answers that are presented in the demonstrator.

The last phase of this approach is devoted to the implementation of the *matching module* that is used to *evaluate* the used algorithms. Based on this evaluation and on the total coverage of this research, this last phase also provides the answers for the research questions stated in section 1.2.1.

1.4 Thesis overview

The structure of this thesis largely coincides with the approach phases as described in the previous section. First, chapter 2 investigates which question & answer properties can be identified and which features play a role in generating congruent question & answer pairs. It provides different perspectives for questions and answers and clearly identifies the "topic-comment" pattern as the underlying pattern for answers to generate. Interesting, in relation to this, is the possible usage of aggregation; What aggregation is, what types of aggregation exist and which form of aggregation is used in this research, is discussed in chapter 3. Chapter 4, describes the experiment that tests the found features and provides the foundation for practical usage in the demonstrator to implement. Next, chapter 5 explores how dependency trees can provide the necessary properties in such a way that the identified features and patterns can be realized. Further, chapter 6 describes the design & implementation of the implemented module and also covers the main revision algorithm responsible for generating the "topic-comment" pattern inside a generated answer. Chapter 7 covers the assessment of the developed module and evaluates the generated answers. It defines the scenario's where the algorithm performs in a reasonable way, but also identifies the scenario's where it performs below expectations. The thesis closes with conclusions and starting points for future work in chapter 8.

Chapter 2

Q & A matching

2.1 Introduction

A question and an answer form a pair that is semantically and syntactically related. Therefore the assumption is that given a particular question, there will probably exist a certain set of schemas in which a correct answer for that question can be formulated and presented. Identifying these patterns of questions and answers will not only benefit the extraction of text fragments that contain relevant information but will also be useful to revise a text in a way to be presented as a natural answer [Hovy et al., 2002, Negri et al., 2003]. Which question & answer properties can be identified and which features play a role in generating congruent question-answer pairs? is therefore the question to answer and will be provided in this chapter.

2.2 Questions

In common, the term *question* can be used in different ways [Stokhof and Groenendijk, 1997] and be looked at through a three layer perspective. The term may, first of all be used to refer to a particular type of sentences, characterized by word order, intonation, question mark and the occurrence of interrogative pronouns. This layer can best be described as the "syntax-perspective" of a question based on the grammar of the language in which the question is constructed. This type of sentences can be referred to as *interrogative sentences*. The term *question* can also refer to the speech act that is typically performed in speaking interrogative sentences. For example, to denote a request to an addressee to provide certain information, a request to answer the question. Such speech acts can be referred to as *interrogative acts* and can be described as the act of asking a question. This layer can best be described as the "pragmatical-perspective". A third use of the term *question* is also possible when trying to identify the meaning of a question based on semantics. Different approaches are available for extracting the meaning of a question (see section 2.2.1). These approaches are based on the translation of semantics to a logical representation and the global view can be placed in the third layer, the "semantical-perspective'. It is best described as the the *sense of an interrogative act*.

For this research the first definition of a *question*, as seen from the "syntactical-view", can be used to identify patterns and question types purely on the pattern of the question. But this is not enough when looking at the whole picture of a QA-system. "What" is being asked about is as essential as the syntactical question used to formulate it. In order to identify the meaning of questions and to identify what is asked "about" in questions, a theoretical foundation can be provided, based on the *sense of an interrogative act*.

2.2.1 Sense of an interrogative act

In the logical, philosophical and linguistic literature, a number of theoretical frameworks have been proposed for the meaning of questions (see [Stokhof and Groenendijk, 1997] for an overview). Two of these, that figured prominently in linguistic semantics are called the **proposition set approach** and the **structured meaning approach**, [Krifka, 2001]. A description of these approaches shows the formal ideas regarding questions and the way corresponding *congruent answers*[1] are identified. That is, the meaning of a question is traced back to the meaning of the answer and could provide rules for answer patterns related to questions. It supports the idea of strong relations between a question and an answer.

The Structured Meaning Approach

The structured meaning approach goes back to Ajdukiewicz (1928), as noticed in [Hiz, 1978]. It was developed by Hull and got further attention of Tichy, Hausser & Zaefferer, von Stechow, Zimmermann, and Ginzburg. By generalizing over a number of important differences between the theories that follow this approach, the basic idea can be described in the following way: *"Question meanings are functions that, when applied to the meaning of the answer, yield a proposition".*

The example in (1) illustrates the above. The meaning of the sentence *"Who read Silence of the lambs"* is a function that, when applied to the meaning of the answer, here *"John"*, gives a proposition. The proposition which is in this case the meaning of *"John read Silence of the lambs"*. By using a standard version of truth-theoretical interpretation, the meaning representation $READ(SL)(J)$, yields the proposition.

(1) a: Who read *Silence of the lambs?* $\lambda x[READ(SL)(x)]$
b: John J
Question applied to answer: $\lambda x[READ(SL)(x)](J) = READ(SL)(J)$

As can be seen in example (1), the used notation does not incorporate the meaning of the *wh*-word itself. There is no distinction, for example, between "Who" and "What". To indicate the function domain of the question meanings; the set of persons for "who" and the set of things for "what", [Krifka, 2001] uses an explicit notation to overcome this problem. It describes the function and its domain as a pair in which the first part of the question meaning forms the **background** and the second part the **restriction** of the meaning. This way, see (2), it becomes clear that an answer is already restricted to a specific set. In other words, it provides a more or less suggested answer type.

(2) a. Who did John see? $\langle \lambda x[SAW(x)(J)], PERSON \rangle$
b. What did John see? $\langle \lambda x[SAW(x)(J)], THING \rangle$

These examples (1 & 2) show the basics of the structured meaning approach and by interpreting these logical representations, the meaning and suggested answer type of a question becomes clear. However, these examples make use of **term answers**, as described by [von Stechow and Zimmerman, 1984], and not of **full answers** that are the answers that are based on the "topic-comment" pattern.

A **full answer** is closely related to the **term answer** and forms the foundation for the proposition set approach, about the meaning of questions. A full answer can be described as: *"The meaning of*

[1] A term coined by [von Stechow and Zimmerman, 1984] and easy to define for simple constituent questions; an answer that fills in a constituent for the wh-word in the question, and does nothing more than that. In other cases it cannot be defined as easily.

a full answer is the question meaning applied to the term answers". This means that the answer for "Who did John see?" will not simply be "Mary", but will be "John saw Mary". This way, the answer provides us with a completer answer that contains more information and is less ambiguous.

The Proposition Set Approach

The proposition set approach goes back to Hamblin (1958) and was further developed and refined by Karttunen and Groenendijk & Stokhof. The essential idea is that the meaning of a question is the set of its possible full answers [Stokhof and Groenendijk, 1997], a so called set of propositions. This is illustrated in example (3).

(3) a: Who read *Silence of the lambs?* $\{p|\exists x[PERSON(x) \wedge p = READ(SL)(x)]\} = \{READ(SL)(x)|PERSON(x)\} = \{READ(SL)(M), READ(SL)(J), \ldots\}$

b: John read *Silence of the lambs.* $READ(SL)(J)$

The answer type of the proposition set analysis is a full answer. In *"Who read Silence of the Lambs? — John"*, the answer has to be spelled out first as *"John read Silence of the Lambs"* before it can be interpreted. This interpretation takes place by some syntactic process[2] that deletes material recoverable from the question, in other words: the topic. What this topic exactly is and how to look at it, will be handled in section 2.3.2.

(4) a: Who [read *Silence of the Lambs?*]$_A$

b: John $\emptyset_A$

Spelling out: John [read *Silence of the Lambs*]$_A$

While the proposition set approach in this section has an other point of view for the meaning of questions in comparison with the structured meaning approach, it also illustrates that there exists more than just one answer for a question. Purely based on logic rules the theory is capable of translating a question & answer pair in a set of propositions to understand the question meaning and to identify a possible answer. For other examples and a deeper overview of the theory, see [Krifka, 2001], since this section only introduced some of the basic principles to get an idea of what questions are and how they can be semantically analyzed.

Summarizing

Both these theoretical frameworks regarding the meaning of questions clearly illustrate that there exist relations between questions and congruent answers in terms of logical representations. The answers can be looked at as 'term' or 'full' answers, in order to understand the meaning of a question. The theories provide logical representations for questions in order to understand their meaning, but translating these logical rules into syntactical patterns seems not practical. This is because the theories try to come up with a way of extracting the meaning of questions and don't say enough about the syntactic structure of a question. Looking at the word order in a question and using dependency relations from the question structure, can provide insight into the syntactical structure of questions. However, for the identification of question patterns, the "syntactical view" of questions seems a better approach.

[2]Rarely discussed by people that support the proposition set analysis

2.2.2 Syntactic structure

In general there are three commonly known **basic** question types and this also counts for Dutch, (See ANS [Haeseryn et al., 1997] Algemene Nederlandse Spraakkunst for information regarding the Dutch language). There are "yes/no"-questions, "choice"-questions and "question word"-questions[3] each with their own specific sub-classes. A description of a "yes/no" question is very obvious, it expects a 'yes' or 'no' as an answer and mostly starts with an finite verb. For example *"Komt hij eten?"* ("Is he coming for dinner?") and *"Is RSI te genezen?"* ("Can RSI be cured?") are "yes/no"-questions. A "choice" question is harder to describe, although an answer can more or less be extracted from the question itself. This because the possible answers are indicated inside the question and separated by the word 'or', 'of' in Dutch. For example, *"Wil je koffie of thee?"* ("Do you want coffee or tea?") can be identified as a "choice"-question. However, as will be explained later on, there are some difficult issues related to "choice" questions. Most interesting are the "wh"-questions, because these can have a lot of different forms. *"Hoeveel geld heb je nog?"* ("How much money do you still have?") and *"Welke symptomen heeft RSI?"* ("Which symptoms does RSI have?") are two examples of "wh"-questions. The addition of other words to these wh-words can be used to inquire about specific information. For example, by using the word "often" to construct "How often?" the question will inquire specifically about the frequency of some event.

More complicated question structures are possible through an integration of questions in one complex sentence, so called "multiple constituent" questions (for example: "Who read what?"). Or by using two question types in one sentence (for example: "If RSI is not a disease, then what is it?"). Such questions are not covered by the "simple" questions statement and won't be investigated in this research. To know what questions are being investigated in this research, lets take a look at the formal structures that are described below.

"Yes/no"-questions

The description of a "yes/no"-question is simple, it expects a 'yes' or 'no' as an answer and the question structure usually starts with a finite verb (in Dutch a so called "voor-pv"). Although they usually start with a finite verb, it is also possible to create a "yes/no"-question by a statement sentence. For example the following question *"De tiende is het vandaag?"* ("The tenth it is today?") can only be identified as a question by intonation or by the question-mark at the end of the sentence. It cannot be identified as a question by identification of the finite verb as first part of the sentence. Although this form of a "yes/no"-question can be translated into the formal format for this type of question, see table 2.1, it remains doubtful whether people use this kind of sentences, when using a QA-system.

Question type	**Question structure**
Yes/No	[finite verb] **subject** ... ?
	"Kan RSI worden voorkomen?", "Kan rsi verdwijnen?"
	"Can RSI be prevented?", "Can RSI disappear?"

Table 2.1: "yes/no"-question structure

The formal definition in table 2.1 is very general, it identifies the finite verb on the front together with a subject and the remainder of the question. Analysis of the build corpus should provide insight whether this formal structure is broad enough to make distinctions between question types. Also worth mentioning is that through the use of words like "niet, geen, niets, wel, etc"[4] in a "yes/no"-question, a suggestion is given for the answer. See (A) for some examples that illustrate this. This suggestion for the answer is usually supported by context/situation and it is doubtful if

[3] "Question word" questions are also known as wh-word questions and this type of questions will therefore, further be referred to by 'wh-questions'

[4] "Immers, toch and zeker", in the case of a statement sentence

these forms of questions will be used in a QA-system. This, since it is commonly used in dialogs, but not in written text.

(A) "Bent u *geen* docent geweest?" (Expected answer: 'yes', 'ja' in Dutch)
"Heb je de brief *wel* gepost?" (Expected answer: 'no', 'nee' in Dutch)
"Het licht is *zeker* uitgevallen?" (Expected answer: 'yes', 'ja' in Dutch)

Choice-questions

A "choice"-question is complex, but the answer can more or less be extracted from the question sentence itself: the possible choices, or alternatives, are mentioned inside the question itself, separated by the word 'or' ('of' in Dutch). See table 2.2 for an example. Based on this word 'or', it looks like it is relatively easy to identify this type of question. However, there are some remarks on this assumption.

Question type	**Question structure**
Choice	[finite verb] **subject** ... [OF] ... ?
	"Is een stijve nek wel of niet een symtoom van RSI?"
	"Is a stiff neck a symptom of RSI yes or no?"

Table 2.2: "choice"-question structure

A question like *"Wilt u liever koffie of thee?"* ("Do you prefer coffee or tea?") is a "choice"-question if coffee and tea form a distributive. In other words, the question can be transformed to: *"Wilt u liever koffie of wilt u liever thee?"* ("Do you prefer coffee or do you prefer tea"). It can also represent a "yes/no"-question if coffee and tea form a collective, in this example the collective of warm drinks. So dependable on the context of the question, the type of the question can change. What we have now is that sometimes "choice"-questions cannot be distinguished from "yes/no"-questions based purely on the sentence structure. However, by identifying the word 'or' it is possible to make a distinction. And besides, this situation will only occur if a questioner already has specific knowledge about a domain. Otherwise it is almost impossible to come up with such questions. Therefore it is possible to state that: if there exist a 'of' inside a question structure, the question sentence can be marked as a "choice"-question. With this statement in mind, the example question "Do you prefer coffee or tea?" will be identified as a "choice"-question. Also the question "Is a stiff neck a symptom of RSI yes or no?" will be identified as a "choice"-question, although the alternatives in this question are represented by the 'yes' & 'no' words.

"Question word" questions

The "wh"-questions are the most interesting questions, because these can have a lot of different forms and there exists a lot of variation in the wh-words. An interesting property of the wh-words is that the identification process of an expected answer type, see table 2.3, is almost straight forward. This variation in wh-words also makes it a difficult type when it comes to question meaning and identification of underlying patterns. Usually a "wh"-question starts with a wh-word in front of the question sentence, but as will be described in this section, other variations are possible.

One property of a couple of wh-words is the possibility to inquire about specific information. This is possible through the use of specific "question words" or the addition of an extra word, see table 2.4. This is important for pattern identification, because an extra word needs to be identified. Further information regarding this suggested answer type and the usage of semantic comparison is provided in section 2.3.

Question Word	In Dutch	Expected information
When?	Wanneer?	Time
Where?	Waar?	Place
Who?	Wie?	Person
Why?	Waarom?	Reason
How?	Hoe?	Manner
What?	Wat?	Object/Idea/Action

Table 2.3: Question words and expected information

More specific	In Dutch	Expected information
Which (one)?	Welke?	Choice of alternatives
Whose?	Wiens?	Possession
Whom?	Wie?	Person(objective formal)
How much?	Hoeveel?	Amount (non-count)
How many?	Hoeveel?	Quantity (count)
How long?	Hoe lang?	Duration
How often?	Hoe vaak?	Frequency
How far?	Hoe ver?	Distance
What kind (of)?	Welke soort	Description

Table 2.4: Words to inquire more specific information

So the "wh-word" identification already tells us a lot. For sure that it is a "wh"-question, but it also suggests the expected answer type. A formal specification for the "wh"-questions structure can therefore strongly rely on the occurrence of the "wh-word" in the question pattern, see table 2.5.

Question type	Question structure
Question Word Question	[question word] [verb] **subject** ...
	"Wie kan RSI krijgen?, Waar kan rsi ontstaan?
	"Who can get RSI?", "Where can RSI arise?"

Table 2.5: "Question word" question structure

The structure in table 2.5 covers the basic pattern for "wh"-questions that can be stated as "simple" questions. A broader structure is hard to identify, certainly when we look closely at some possible variations in sentence structure in (B).

(B)	"Met *wie* heb je gesproken?"	"To *whom* did you speak?"
	"Van *welke* professor krijg je commentaar?"	"From *which* teacher do you get comments?"
	"*Hoeveel* geld heb je nog over?"	"*How many* money do you still have?"
	"RSI wordt veroorzaakt door *wat*?"	"RSI is caused by *what*?"
	"*Waardoor* ontstaat RSI?"	"*Through what* arises RSI?"

To come up with a general structure, that can identify all the possible formats of a "wh"-question, would be ideal. However, since the collection of possible "wh"-questions that can be formulated is huge, and therefore also the set of possible patterns, it is better to look at the "real-world" corpus. In order to provide a pattern that is practical in use for this research, an analysis of the corpus is necessary. This analysis, section 2.2.3, identifies the patterns available in the corpus and provides an overview regarding "wh"-, "yes/no"- & "choice"-questions.

2.2.3 Corpus based structure

Since there is no working Q&A-system for Dutch it is necessary to build a corpus consisting of questions and text fragments that contain an answer for that specific question. It also provides valuable information regarding the frequency of each question type used and can provide relevant information about the question patterns. The latter is especially important for this section, since it provides "real-world" patterns regarding the medical RSI domain.

The corpus

The creation of the corpus is based on "real-world" questions extracted from web-sites related to Repetitive Strain Injuries (the RSI domain). Some of these web-sites[5], contain pages with questions and accompanying paragraphs that function as answers. Out of these paragraphs it is possible to select sentences that by itself, can function as retrieved text fragments in the corpus. Other sites[6], provide general information or online discussion boards where questions are asked regarding the RSI domain. By using questions that can be extracted from discussion boards on internet or from RSI related sites, the corpus can be seen as a"real-world" corpus. Of course, it still forms a small amount of questions that can be asked regarding the RSI problem, but it can be seen as representative for questions asked by people on internet. At least, representative for simple questions as defined in section 1.2.1.

So the questions and fragments are extracted from material that can be found on internet. An extra constraint while creating the corpus, is the coverage of the identified question types: "yes/no", "choice" and "wh"-questions. Extracting the questions from the internet resulted in 6 "yes/no"-questions, 0 "choice"-questions and 25 "wh"-questions. As it turned out, zero "choice"-questions were covered inside the corpus. Not because we were not looking for this type of question, but because accompanying fragments were hard to find. This is probably because to formulate a question with alternatives, it is necessary to have specific knowledge about the RSI domain and not that many people seem to have that knowledge. Besides, it was hard to find "choice"-questions, it was impossible to collect an accompanying text fragment for this type of question. However, based on the word 'or', this questions type can still be identified. The "yes/no" and "wh"-questions are both contained in the corpus but each with a different frequency. The 6 "yes/no"-questions may seem less compared to the 25 "wh"-questions, but this is to cover the variation in wh-words.

To use all this corpus material for the identification of question structures, all the material has been annotated by the Alpino parser[7]. This annotation process resulted in question sentences and text fragments with POS-tagging, surface patterns and dependency information, all stored inside XML files. Because there are no "choice"-questions available in the corpus, the focus lays on the "yes/no" and the "wh"-questions to confirm the structures as described in table 2.1 and table 2.5.

[5]For example www.rsi-vereniging.nl and www.arbeid.tno.nl/kennisgebieden/rsi
[6]For example www.medicuisine.nl and rsi.pagina.nl
[7]Section 5.3 provides information about the Alpino parser; what it is and what it is capable of.

Identified patterns

The question patterns of the "yes/no"-questions and "wh"-questions in the corpus are based on the identification of POS-tags and categorial dependency relations, relations like NP (Noun phrase) & PP (prepositional phrase). The identified patterns can be found in table 2.6 and in table 2.7 for respectively the "yes/no"- and the "wh"-questions.

Question type	Corpus structure	Frequency
Yes/no	[verb] **subject** . . . ?	
	([verb] [adv] [noun] ([prep] [name])pp)sv1 ?	1
	([verb] [noun] ([prep] [name])pp)sv1 ?	1
	([verb] [name] [fixed] ([prep] [name])pp)sv1 ?	1
	([verb] [noun] [name] [verb])sv1 ?	1
	([verb] [name] [verb])sv1 ?	2
Total		6

Table 2.6: "Yes/no" corpus structure

The theoretical assumption for the identification of a "yes/no"-question was based on the pattern "*[finite verb]* ***subject*** *. . . ?*" and corpus analysis reveals that this pattern is sufficient for question type identification within the used corpus. Since Alpino is able to identify POS-tags, the surface patterns in the corpus can be matched with the theoretical identified patterns attached to each question type. A remark is that the POS-tags that Alpino delivers, cannot be divided into specific tags. For example, splitting 'verb' into 'finite verb' and 'auxiliary verb' tags. Another interesting point is the *sv1*-relation; it stand for a sentence that starts with a verb. So based on this small corpus analysis it can be stated that all the "yes/no"-questions use a verb on the first spot in a sentence.

Question type	Corpus structure	Frequency
"Wh-word"	[question word] ([verb] / [noun]) . . . **subject** . . . ?	
	([WH-word] [verb] [name])whq ?	5
	([WH-word] [verb] ([noun] ([prep] [name])pp)np)whq ?	4
	([WH-word] [verb] . . . ([noun]/[name]))whq ?	4
	([wh-word] [verb] [name] [verb])whq ?	4
	([wh-word] [noun] [verb] . . .)whq ?	4
	([wh-word] [verb] [name] [noun])whq ?	1
	([wh-word] [verb] [pron] [name])whq ?	1
	([wh-word] [adj] [verb] [name])whq ?	1
	([wh-word] [noun] [prep] [name] [verb] [adv])whq ?	1
Total		25

Table 2.7: "wh" corpus structure

The theoretical assumption for the identification of a "wh"-question was based on the pattern "*[question word] [verb] . . .* ***subject*** *. . . ?*". It identifies the wh-word and the subject of the question sentence. When looking at the corpus it becomes clear that there exist a lot of patterns that can be substituted for the '. . . ' in the theoretical pattern. The variation of the second word is the most important point of interest. Some "wh"-questions (starting with 'Welke' and 'Hoeveel'), seem to have a 'noun' as second word in the pattern, while other "wh"-questions (starting with 'Waar', 'Wie', 'Wanneer', 'Waardoor', etc.) all seem to have a 'verb' as the second word. Because of this difference, it seems a good approach to handle both pattern situations differently. Identifying a pattern for questions that have a 'noun' at the second spot in the sentence and a pattern for questions that have a 'verb' on the second spot in the sentence. These assumptions result in a "*[question word] [verb] . . . [subject] . . . ?*" pattern and a "*[question word] [noun] . . . [subject] . . . ?*"

pattern. Then, further analysis of the wh-word itself, provides enough information to give a formal structure for the accompanying answer. In section 2.3, the further analysis of the wh-word gets more attention.

2.2.4 Information need

Now that it is known what questions are, how they can be interpreted and what kind of question types there exist it is time to look back at the use of the term *question*. As described in section 2.2 there exists a three layer perspective for questions and the "semantical" and "syntactical" perspectives have been investigated at this point. However, the third perspective, the "pragmatical-perspective" has not been discussed further although it has influence on the use of a question.

In [Campbell, 2000] the issue "information need" is explored, because the need for information changes in time when people use information retrieval systems. Campbell tries to come up with a model to cover the change in information needs and to be able provide a solution for this issue in future information retrieval systems. Change in information need can best be described by the following example:

> First, there is an 'information need' that is abstract and for example based on the the word "car", meaning that someone has is looking for information of cars. Accordingly this 'information need' has to be formulated in a question like "What is a car?". When the information system provides an answer for this question, the 'information need' might be satisfied, but is likely to change towards a more specific question like for example "What are sport cars?". As more search attempts occur in time, the 'information need' evolves and might in the end not reflect the original 'information need' at all.

This phenomenon is interesting, because it has influence on the way question sentences are used to ask for information. They can be used the express a vague topic that is based on knowledge that only the person asking the question has, or question can be used within a shared knowledge domain. This is the case in a dialog act, where there shared information can be extracted out of the dialog. The assumption for this research is that questions are asked to retrieve information about a certain topic that the user doesn't know yet. So the questions investigated in this research are restricted to a context that can be seen as shared knowledge for the questions that can be asked. In this research this context is the RSI domain and the system is being used to retrieve information that is no common knowledge of the user, but is knowledge of the QA-system. However, a user might also use a question to test the consistency of the system in presenting an answer. In other words, is the system able to correctly answer a question of which the user already knows the correct answer? These kind of questions are not covered in this research, because at this stage there is no working retrieval agent. However, in the future it might be a way of testing the performance of a finished QA-system.

2.2.5 Formulations & Ambiguity

Formulations

Different formulations (a difference in lexical choice or syntactical variation) can be used to ask for the same information. For example, *"Is het mogelijk om RSI te behandelen?"* ("Is it possible to treat RSI?") and *"Kan RSI worden behandeld?"* ("Can RSI be treated?"). These two questions can both be identified as a "yes/no questions" and both can be answered by a 'yes' or a 'no'. Hence, comparing the questions on a truth-conditional basis, results in equivalent questions. So the answer to present can also be truth-conditional equivalent. However, when it comes to providing

information out of the original question (realizing an "topic - comment" pattern as discussed in section 2.3), the used words will be different. For the above examples, the formulations *"Ja, het is mogelijk om RSI te behandelen ..."* ("Yes, it is possible to treat RSI ...") and *"Ja, RSI kan behandeld worden ..."* ("Yes, RSI can be treated") are likely to form the initial starts for an answer. Both these phrases are extracted from the accompanying questions and form the foundation for the presentation of a generated answer. Another example, *"Is het mogelijk om RSI te behandelen of is het ongeneeslijk?"* and *"Bestaat er een behandeling voor RSI?"* illustrates syntactical variation. Both questions ask for the same information, but in the fist a "choice"-question can be identified and in the second a "yes/no"-questions can be identified. However, both questions ask for a answer with the type boolean. The assumption is that different formulations have effect on the presentation of answer text, but this issue will be tested in the experiment described in chapter 4.

Ambiguity

Besides different formulations there is also the problem of ambiguity within questions. Certainly when there doesn't exist any information about the question context. The question *"Gaat John met de fiets naar de stad?"* ("Is John going by bike to the city?"), can be interpreted in different ways. In the following overview, the *italic* parts in the question sentences represent the intonation within the question.

Question:	Context:
a. "Is *John* going by bike to the city?"	"Who is going by bike to the city".
b. "Is John going *by bike* to the city?"	"How John is going to the city"
c. "Is John going by bike *to the city*?"	"Where John is going by bike"
d. "Is John going *by bike to the city*?"	"How and where is John going"
e. "Is *John going by bike to the city*?"	"Who, how and where is someone going"

As these examples clearly illustrate, the context (and such the 'information need' plan behind it) is important to decide "what" can be seen as the thought behind the formulated question. A definition for this so called "question topic" will be provided in the next section. Since the context in this research is fixed to the RSI domain and QA-systems provide people the opportunity to ask something regarding that domain, the broadest context (comparable with option e. in the above example) imaginable is a logical choice to be used for the 'question topic' definition.

2.3 Congruent answers

What we have now is a clear overview of what questions are and what properties and issues (meanings, formulations, types, patterns, etc) there exist to identify. As mentioned in the previous, a question suggests a lot about the kind of information expected in an answer. However, it looks like it suggests almost nothing about the syntactic pattern of presenting that information. There are studies related to information structure, for example the **centering theory** [Grosz et al., 1995] and **information packaging** [Vallduvi, 1993, Kuboň, 1999] provide a formal theory of the notions of given and new information, amenable to computational implementation. Although, the centering theory provides insight into focus of attention, into coherence of utterances within a discourse and into syntactic patterns, this section is focused on information packaging. Specifically, the topic-comment pattern is explained and identified as a syntactical pattern for the presentation of answer texts.

2.3.1 Information Structure

Alternative sentential structures, differing in word order, in intonation structure, or in both, may be used to express the same propositional content. Despite their truth-conditional equivalence, these sentential structures are not equivalent in absolute terms. Several pragmatic components have been argued to affect sentential form (illocution[8], empathy[9], etc.) but the component that has been studied the most in its relationship to syntactic form is "information packaging" [Vallduvi, 1993]. It can be described as: when communicating a proposition p a given speaker (or output system) may encode p in different sentential structures according to his/her beliefs about the hearer's knowledge and attentional state with respect to p. The informational primitives that underlie this diversity in sentential form are not that clear, but the terms *focus, (back)ground, topic, comment, given, new, theme,* and *rheme*, among others, all try to indicate these primitives.

Informational primitives

The information packaging primitives proposed in the literature share one characteristic: they divide a sentence into a part that anchors the sentence to the previous discourse (in this case a question) or the hearer's 'mental world' and an informative part that makes some contribution (in this case the actual answer) to the discourse or the hearer's 'mental world'. The differences between them lie mostly in the way the primitive notions used are defined. Despite the differences, the proposals can be reduced, based on their empirical predictions, to only two: (A) those that divide the sentence into `GROUND` & `FOCUS` , and (B) those that divide the sentence into `TOPIC` & `COMMENT`.

Ground-focus The ground-focus relation divides the sentence into a noninformative, known, or expected part, the *ground*, and an informative, newsy or dominant one, the *focus*. A single proposition may be 'packaged' in several different ways in terms of the ground-focus partition[10]. This is illustrated in (1), an example taken from [Vallduvi, 1993], where F-labelled brackets are used to delimit the focus and small caps indicate the lexical item associated with focal prominence.

(1) a. "What about the pipes? In what condition are they?"
"The pipes are [$_F$ RUSTY]."
b. "What about the pipes? What's wrong with them?"
"The pipes [$_F$ are RUSTY]."
c. "Why does the water from the tap come out brown?"
"[$_F$ The PIPES are rusty]."
d. "I have some rust remover You have any rusty things?"
"[$_F$ The PIPES] are rusty."
e. "I wonder whether the pipes are rusty."
"The pipes [$_F$ ARE] rusty."

As illustrated, each one of the sentences with a different ground and focus provides an answer to questions with a different context in mind. This to highlight that questions asked with a different contexts in mind can be answered by the same sentence structure. For flat text it might be a possibility to use boldface / italic or to give the focus a special place in the answer by intonational primitives. Another option is the use of a different syntactical patterns like topic-comment to provide extra attention to the focus of the answer, something that seems more interesting when looking for syntactical answer patterns.

[8]The effect a speaker wants to achieve in making the utterance.

[9]The ability to imagine oneself in another's place and understand the other's feelings, desires, ideas, and actions.

[10]The ground has also been called 'background', 'presupposition', 'open-proposition', 'oldinfo', 'given', 'theme' and 'topic'. The focus has also been called 'new', 'newinfo', 'rheme' and 'dominant constituent'. This to give an impression about all the different terms that, in the end, try to explain the same phenomenon.

Topic-comment A characteristic definition for the topic-comment articulation is based on Gundel (1988):

> An entity, E, is the topic of a sentence, S, if in using S the speaker intends to increase the addressee's knowledge about, request information about, or otherwise get the addressee to act with respect to E.

The topic has also been described as a 'point of departure for the clause as a message' or as a 'reference frame' for the sentence. It is clear that *topic* performs the anchoring role to the previous discourse (in this case a question) or the hearer's mental world, while the comment is what makes a new contribution (in this case the answer). The examples in (2) illustrate three different topic assignments on a truth-conditionally equivalent sentence, where the topic phrase appears in boldface.

(2) a. "What about **John**?"
"**John** zag de wedstrijd gisteren."
"**John** saw the play yesterday."
b. "What about **yesterday**."
"**Gisteren** zag John de wedstrijd.'
"**Yesterday** John saw the play."
c. "What about **the play**?"
"**De wedstrijd** die John gisteren zag.'
"**The play** John saw yesterday."

These three sentences in (2) all form a good answer for the question *"What about x?"*, where x stands for the appropriate topic phrase. As becomes also evident, there seems to be a correlation between topichood and lefthand position (lefthand placement, or left-right-principle in Dutch) in the clause. Some authors, [Halliday, 1985] for example, even claim that topics always have to be sentence initial. This correlation, of topichood and lefthand placement, is interesting when trying to identify a pattern that forms the foundation for congruent answers. Lets take a closer look at this "topic - comment" or lefthand placement phenomenon in detail.

2.3.2 Topic - Comment

When using a topic-comment order, as pattern for the generation of an answer, the comment functions as the new contributing information for the topic that is repeated at the start of the answer. In other words, the topic-comment order first repeats the already known information and then provides the new contributing information that is asked about. In order to realize this pattern however, it is necessary to provide clear definitions for the terms 'topic' and 'comment'. Hence, it is also necessary to provide definitions for the terms 'question topic', 'answer topic', 'answer comment' as these are used construct an answer based on the topic-comment pattern.

Topic The topic provides the anchoring role to the previous discourse and usually contains old, already known information.

Comment The comment is what expresses the new contributing information about the topic.

These two definitions are very global and abstract and need specification when used specifically for congruent question & answer pairs. Especially an explanation of the term topic in general is necessary, since this is the most frequently used, unexplained, term in discourse analysis literature as explained in [Lin, 1997].

The most important distinction of topic is the one between *discourse topic* (what a part of a discourse is about) and *sentence topic* (what is predicated about an entity in a sentence). Besides that, topic is also described as the old (given) information of a sentence. The following example illustrates this:

(Example)	Mr. Johnsen is a careful researcher and a knowledgeable linguist, but his originality leaves something to be desired.
Sentence topic:	Mr. Johnson.
Discourse topic:	Mr. Johnson's abilities.

Related to Q&A matching

When relating all this to question & answer pairs it becomes clear that the 'question topic' can be compared with the problem of finding a 'discourse topic'. So for this research, the question topic can be used to link the new contributing information, that forms the actual answer, to the question. Knowing this, it is possible to provide definitions for the following terms as used in this research:

Question Topic The thing someone intends to increase his/her knowledge about by using the question.

Answer Topic A repetition of the question topic into the generated answer to link the answer to the question.

Answer Comment The information in the answer, that expresses the actual information asked for in the question.

By realizing this topic-comment pattern in the presentation of answer text, the relation between word order and informative importance of a sentence can be described as: in sentences, the informative less important elements (topic, theme) occur before the elements with a more important informative value (comment, rheme). In other words [Haeseryn et al., 1997], there is a kind of rising line in a sentence: what is informative important occurs more to the back of the sentence.

In Dutch, like in Swedish and a number of other Germanic languages, an additional alternative is available: the configuration called 'contrastive dislocation' (Zaenen). It is a fronting operation that binds a pronominal which appears in a lefthand slot, but to the right of the contrastive-dislocated phrase. Sentence (4)a contains a contrastive-dislocated link, while (4)b illustrates a topicalized link. In (4)c the same link appears in its unmarked order.

(4) a. "**De vijf laatste films van Godard**$_1$ die$_1$ heeft Jan gezien."
"the five last movies of Godard that has Jan seen."
b. "**De vijf laatste films van Godard** heeft Jan gezien."
c. "Jan heeft **de vijf laatste films van Godard** gezien."
"the last five films by Goddard Jan has seen."

The difference between (4)a and (4)b is the usage of the word 'die' and provides **De vijf laatste films van Godard** with an extra reference and as such is indicated as important, as new contributing information.

So in general, by placing the topic of the previous discourse (the question in this research) at the front of an answer, it provides a good starting point for a congruent answer. It first repeats what is already known and therefore less important (topic), then it provides the information that is actually asked about (comment).

Concluding

The packaging of information provides extra importance to specific parts of information. Parts that can be extracted from the question (the defined 'question topic'), can be repeated at the start of a following discourse and as such realize a congruent answer based on the topic-comment pattern. This is accomplished by placing the 'question topic' in front of the information that provides the actual answer. In other words, by repeating information out of the question to provide a clear context, in which the contributing information can be placed. When taking this pattern for granted, an issue that remains is the following; can different types of answers be identified and do these have influence on the topic-comment pattern? The next section provides insight into this issue.

2.3.3 Answer surface patterns

It would be practical for the presentation of answer, if suggested answer types have fixed surface patterns. Because, then it would be a matter of filling in the actual fact within the answer pattern. It is already known (based on the information structure theories section 2.3.1) that old but still relevant information usually occurs at the start of an informative sentence and the new contributing information in the successive part of that sentence. But, what kind of effect do the three basic question types have on this pattern. For each question type the suggested answer types are used to identify these pattern.

"Yes/no" questions The identification of this type of question is described in section 2.2.2 and the expected answer type is simple stated 'yes' or 'no'. For a reliable answer it is preferred to provide background information [Lin et al., 2003]. By providing extra information, so called background information to make the context of the answer clear, it functions as a confirmation for the user that the system provided an answer for the original question asked. According to the topic-comment pattern, the 'topic' of a question like "Kan RSI worden voorkomen"? ("Can RSI be prevented?") needs to be repeated before the new contributing information is provided. A natural pattern would be: *"[yes/no] [topic of question] [background information]"* and to test this pattern, the experiment in chapter 4 will test the users preferences for presenting the answer for a "yes/no"-question.

"Choice" questions The alternatives provided in a "choice"-question suggest the answers. Like the answers for "yes/no"-questions, answers for "choice"-questions also need background information to provide a confirmation for the user that the system provided an alternatives out of the original question asked. Based on the topic-comment pattern once again, a pattern for providing an answer for the "choice"-question type would be *"[topic of question] [an alternative] [background information]"*. The experiment in chapter 4 won't test the users preference for presenting the answer for a "choice"-question, because no such questions with accompanying fragments were found as noted in section 2.2.3.

"Question word" questions For the "wh"-questions, a pattern is not as implicit as for the other two types of questions, because the variation in answer types is bigger. As mentioned in section 2.2.2, the expected answer type depends on the used wh-word. However, these expected types could be similar when realizing a pattern for the presentation of answer texts. The overview below provides a pattern for each suggested answer type accompanied with some examples.

Time/Place/Person - Indicated by the wh-words 'When', 'Where' and 'Who' the expected type of information must be present in the answer. A question like, "Waar kan RSI ontstaan?" ("Where can RSI arise?") is usually answered by a the pattern: "*[topic of question] [the different places where RSI can arise] [extra information]*". Or in other words, by "RSI kan ontstaan ...", ("RSI can arise ..."). However, a question like "Wie kan RSI krijgen?" ("Who can get RSI?") can be answered by "RSI kan gekregen worden door ..." ("RSI can be gotten through ...", this is a straight translation into English) or by "...kunnen RSI krijgen ..." ("...can get RSI ..."). The patterns for these answers are respectively "*[topic of question] [different person(s)/groups] [background information]*" and "*[different person(s)/groups] [topic of question] [background information]*".

Reason/Manner/(Ojbect/Idea/Action) - Again indicated by wh-words, but this time by 'Why', 'How' and 'What' the expected type must be present in the answer. Once again the location where to put this information in a pattern is not that obvious. For the question "Hoe begint RSI?" ("How does RSI begin?") it is possible that 'the manner' in which RSI begins is vague. However such an answer still needs to be covered inside a pattern and this is possible. For example: "*[topic of question] [contributing information (the manner)]*", "RSI kan ontstaan ..." ("RSI can begin ..."). However, for other questions like "Welke factoren leiden tot RSI?" ("Which factors result in RSI?") an answer can also be in the form "... zijn factoren die kunnen leiden tot RSI". In the latter the order of the topic and comment is turned around resulting in the pattern "*[contributing information (the factors)]*".

Amount/Duration/Frequency/Distance - The wh-words that inquire about specific information, behave exactly the same regarding answer structure constraints, as the previous answer types above An answer for "Hoeveel mensen hebben RSI?" ("How many people have RSI?") can be answered by the pattern "*[contributing information (amount)] [topic of question]* or based on information structure with the pattern "*[topic of question] [contributing information (amount)]*".

The identified patterns for the answer types, clearly show that variation is possible. Hence, all variations as discussed above, are grammatical correct, but they conflict with the identified topic-comment pattern out of section 2.3.2. To be able to call the topic-comment pattern a good pattern for the presentation of answer texts, further investigation is necessary by means of an experiment, seeẽfecha:experiment. An experiment to see what the users preferences are for the presentation of an answer. First, I'll present a summary of the features and properties that have been identified in this chapter.

2.4 Matching features & properties

After reading this chapter it is clear that a question is more than just a sentence, characterized by word order, intonation, question mark and the occurrence of interrogative pronouns. It is an interrogative act that can be used to denote a request to an addressee to provide certain information, a request to answer the question. It also has become clear, that answers are restricted to constraints related to answer type, pattern and contents to be able to function as congruent answer. A summary of the found properties and features seems appropriate, not only for summarization, but also for hypotheses to be tested:

Question Types There are three basic different question types that can be identified. "Yes/no", "choice", "wh-word" questions each with their own expected answer type. When focusing on the wh-word questions, it is clear that they can be subdivided by the used wh-word, since this word suggests a lot about the expected answer.

Answer Types For the "yes/no" and "choice" questions the answer types are fixed. A "yes/no"-question can be answered by one of the two booleans 'yes' or 'no' and a "choice"-question can

be answered by on of the alternatives provided in the "choice"-question. For "wh"-questions it all depends on the used wh-word to be able to identify an answer type. However, this wh-word can be identified and each wh-word provides a suggested type.

Question Patterns Per question type there is a pattern that can be identified and used to assign a question type to a sentence. The patterns provide the elements that include the used words choice, the provided alternatives, the dependency relations, etc.

Answer Pattern In general the patterns for answers can be analyzed based on information structure theory. The less informative important elements (topic, theme) occur before the elements with a more important informative value (comment, rheme).

Question Formulation Different formulations can be used to ask for the same information. For example, "Is RSI te behandelen?" ("Is RSI treatable?") and "Kan RSI behandeld worden?" ("Can RSI be treated?"). These two questions are both identified as a "yes/no question" and both can be answered by a 'yes' or a 'no'. However, when it comes to repeating the question topic, the word choice is likely to be different.

Answer Formulation Based on the identification of the properties mentioned above, it is probably possible to formulate an answer that depends on the question pattern, the corresponding question type, the answer type suggested by the question, the corresponding answer pattern, the used formulation in the question and on the informative important elements from a retrieved text fragment.

2.5 Conclusion

Properties of questions and answers, and features to generated congruent question & answer pairs have been identified. Questions can be investigated by a three layer perspective and each perspective has been investigated. Although, investigation is needed regarding these perspectives, this chapter clearly indicates how questions are used in daily life, that questions can be looked at through a three layer perspective, and what kind of question properties are usable for identification in this research. Congruent answers also have been investigated regarding the topic-comment pattern, expected answer types, etc. The topic-comment pattern is available in everyday text, as indicated by the literature and by the information retrieved from the internet. It should therefore also form the foundation for congruent question & answer pairs. By using this pattern it should be clear what exactly is the new contributing information that answers the question and it should also be clear in what context the answer can be placed. For better understanding of this chapter, the important features and properties are presented in an overview and these result in the following four hypotheses to be tested in chapter 4:

Topic-comment linking It is the users preference to link the question to the answer by repeating the 'question topic' in the generated answer.

Topic-comment order The topic-comment pattern for the presentation of answer texts is the users preference. The less informative elements occur before the elements with an hight informative value.

Explicit use of 'yes/no' The explicit use of 'yes/no' in a congruent answer is the users preference for answering "yes/no"-questions.

Formulations The words used in the question must also be the used words in the answer.

Chapter 3

Aggregation

3.1 Introduction

When investigating means of text revision to present natural answers, it is necessary to tackle the relations between question & answer like we did in chapter 2. It is also interesting to look at other means of text revision, that are focused on generating answer that take less effort to read and are easy understand by a user. A research field that tries to cover these issues is indicated by *aggregation*. To begin with an explanation of the term aggregation seems appropriate, especially since it can have different meanings depending on the context it is used in. In the context of this research. it can be described as the process in which two or more linguistic structures are merged together to formulate a new structure. A formal definition cannot be given yet, since there is no agreed upon definition to be found in the literature [Reape and Mellish, 1999]. It attempts to create concise, cohesive, fluent sentences to emulate human linguistic performance. The terms concise, cohesive and fluent are described in [Shaw, 2002] as:

Conciseness A text is more concise if it can convey the same information using fewer words. This involves removing redundant and inferable information. (Example: realizing the sentence "The patient was admitted on Monday and discharged on Friday" out of the two sentences "The patient was admitted on Monday" and "The patient was discharged on Friday")

Cohesion A text is more cohesive if it is more tightly integrated and acts as a whole. Cohesion is a property that makes a text a semantic unit rather than a jumble of unconnected phrases. (The use of pronouns, ellipsis, substitution of generals for specifics and elision of redundant information.)

Fluency A text is more fluent if it flows more quickly and takes less effort to read. Related factors include variety in sentence structure and lexical items, unambiguousness and adhering to communicative convention. (Example: "John ate a large red appel" is more fluent than "John ate a red large apple")

In [Wilkinson, 1995] however, it is noted that aggregation as a process and as a subfield of NLG has some foundational and conceptual difficulties:

> "The small quantity of previous work which deals with aggregation suffers from some conceptual difficulties, including inadequate definition (i.e. as a process which eliminates redundancy), vagueness as to its position in the generation process, and unclear relation to other phenomena."

In an attempt to eliminate these conceptual difficulties and to try to answer the simple question *what is aggregation?*, [Reape and Mellish, 1999] provide an overview of the aggregation field (agreed upon set of goals, types of aggregation, etc) and concludes that there is both a narrow sense and a wide sense of the term aggregation:

X-aggregated is text which contains no multiple non pronominal overt realizations of any propositional content and no overt realizations of content readily inferable or recoverable from the reader's knowledge or the context which are not required to avoid referential ambiguity or to ensure grammaticality.

Narrow sense Aggregation is any process which maps one or more structures into another structure which gives rise to text which is more x-aggregated than would otherwise be the case.

In other words, aggegation requires an "omission or substitution" to be made "that a listener can only resolve by consulting earlier parts of the text" and that "creates a link to an earlier part of the same semantic unit". For example in "Tom en Tim betreden het gebouw. Ze gaan zitten" (Tom and Tim enter the building. They sit down) 'ze' can be resolved back to 'Tom en Tim' by consulting the previous sentence.

Wide sense Aggregation is the combination of two or more linguistic structures into a single linguistic structure which contributes to sentence structuring and construction.

The wide sense can also be described as the "combination" of any two linguistic structures to produce a third more "complex" structure. For example, the meanings of "De auto is hier" (The car is here) and "De auto is blauw" (The car is blue) can be combined to "De blauwe auto is hier" (The blue car is here). It combines the two sentences in a new sentence that contains the information from the two previous sentences in an aggregated way.

By now it should be clear how aggregation can contribute to the NLG field in generating text that takes less effort to read. However, an issue that still remains is *where* and *how* to apply it in the generation process.

3.2 Aggregation in NLG perspective

So the goal of aggregation is clear, but the questions that remain are where and how to apply aggregation within the Natural Language Generation perspective. The consensus architecture of a three-stage pipeline that exists within the NLG-field, was employed in various NLG systems in the late 1990s. This architecture consists of the following stages, as described in [Reiter and Dale, 1997, Shaw, 2002].

- **Content planner:** Selects the information to be conveyed and determines the text structure to convey the selected information. Other common terms for this module include text planner, strategic planner, and macroplanner.
- **Sentence planner:** Selects lexical items and sentence structures to convey the concepts and relations from the content planner. Other common terms for this module include microplanner.
- **Surface realizer:** Transforms a lexicalized linguistic structure into a linearized string. This module was known as tactical planner, but in current analysis, the tactical planner includes both the sentence planner and the surface realizer.

The text structure in the content planner is based on a discourse structure that specifies the sequential ordering between propositions and specifies relationships between them. In this view, a proposition (also known as the message) is the basic unit to be combined and possible relations to use are based on Rhetorical Structure Theory, (one of the most popular discourse structure theories, RST [Mann and Thompson, 1987]). In this theory the relations between propositions are indicated to build a larger discourse structure. The sentence planner takes the discourse structure plan created by the content planner and refines it into a sequence of lexicalized and aggregated propositions for the surface realizer to transform each lexicalized proposition into a sentence. The surface realizer in the end transforms each lexicalized proposition into a sentence. When looking at these stages, the second stage in a NLG- pipeline can be seen as the first phase where it is possible to handle aggregation. However, as the next section will indicate, there exist different types of aggregation and some of them should already be applied earlier in the NLG-pipeline. The type of aggregation for use in this research is "syntactical aggregation", since it is the most common form used and also possible to apply without using the underlying discourse structures as indicated by RST.

3.3 Types of aggregation

Based on the typology as used by [Wilkinson, 1995] and to answer the question "What is aggregation anyway?" [Reape and Mellish, 1999], the following types of aggregation can be identified: conceptual, discourse, semantic, syntactic, lexical and referential aggregation. This typology is based on linguistic type which places the emphasis on levels of linguistic representation and not on where those levels are operated on in a generation system.

Conceptual aggregation typically reduces the number of propositions while increasing the complexity of some conceptual role. For example, the mapping of `ferrari(x), sportcars(y)` into `car(x,y)`. There is a close connection with semantic aggregation and the distinction is described by [Wilkinson, 1995] as: "If we [accept] the distinction between conceptual (language independent) and semantic (language dependent) levels of representation, then the conceptual level of aggregation can be seen as the deepest form of aggregation."

Semantic aggregation is the combination of two or more semantic entities into one. This combination takes place at a level that is abstracted from syntax, but depends on the nature of the language in question [Wilkinson, 1995]. Semantic grouping is an important term and is really about the ordering and bracketing of semantic content. For example, the mapping of the meanings of "Shannon is Chris zus" ("Shannon is Chris' sister") and "Chris is Shannons broer" ("Chris is Shannon's brother") into the meaning "Chris en Shannon zijn broer en zus" ("Chris and Shannon are brother and sister").

Discourse aggregation is defined by [Reape and Mellish, 1999] as any operation which applies to a discourse structure or text plan and maps it to a better structure or plan (according to some metric better). An example would require explanation of terms out of RST and is not relevant for this research. For information regarding this type of aggregation and RST, I'd like to refer to [Reape and Mellish, 1999, Mann and Thompson, 1987].

Syntactic aggregation is the most common form of aggregation and many systems employ simple forms through the use of specific rules. The two most common rules are (1) the subject grouping rule and (2) the predicate grouping rule. For (1) for example, "Jan en Maartje zijn hier" ("Jan and Maartje are here") from "Jan is hier" ("Jan is here") and "Maartje is hier" ("Maartje is here"). An example for (2) is "Jan is groot en sterk" ("Jan is big and strong") from "Jan is groot" ("Jan is big") and "Jan is sterk" ("Jan is strong").

Lexical aggregation can be subdivided into three types: the mapping of more lexical predicates to fewer predicates (`monday(x1),...,friday(x5)` to the predicate *weekdays*), the

mapping of lexical predicates to (fewer) lexical predicates (`monday(x1),...,friday(x5)` to `weekdays(x1,...,x5)`) and the mapping of the predicates *more* and *quick* to the predicate *quicker* are examples to illustrate the possible occurrences of lexical aggregation.

Referential aggregation can be strongly compared with referring expression generation, see [Reiter and Dale, 1997] for example for a better explanation. An example of this kind of aggregation is converting the sentences "Jan is hier. Shannon is hier" ("Jan is here. Shannon is here") into the sentence "Ze zijn hier" ("They are here") with they coreferential with Jan and Shannon.

As mentioned in the above typology, syntactic aggregation is the most common form of aggregation and already used in generation systems. For this research syntactic aggregation will be investigated to be able to generate natural answers based on "clause aggregation", as described by Shaw [Shaw, 2002]. Shaw speaks of *clause aggregation*, because the basic linguistic units to be aggregated are in his opinion clauses. This is a bit in contrast with the term proposition used in text generation systems. The main difference between a "clause" and a "proposition" is that a clause is a linguistic/syntactic concept while a proposition is a semantic concept. So a proposition refers to a relation that represents an event or state in a domain, while the term clause can be used when syntactic issues are involved. Since this research is focused at syntactic concepts, syntactical clause aggregation seems suitable.

3.3.1 Clause aggregation

Clause aggregation operators can be divided in four major categories [Shaw, 2002], that can be linked to Wilkinson's topology: interpretive, referential, syntactic and lexical. From this point forward, syntactical clause aggregation, with its subcategory paratactic aggregation, forms the main subject. Although hypotaxis is also a form of syntactical clause aggregation, it is not covered in this section. In hypotactic aggregation, constructions are not of the same syntactic category, such as a noun phrase modified by a prepositional phrase ("Mary likes John who is a manager"). To cover this form of aggregation is difficult, because it is necessary to identify the used rhetorical relations when applying hypotactic aggregation. Rhetorical information is not available in this research and as such it is impossible to apply hypotactic aggregation. Parataxis, however is the term that refers to constructions in which elements out of the same syntactic category are linked together. Examples of paratactic constructions are coordinations, such as "John en Mary houden van schilderen" ("John and Mary like painting"). For this form of aggregation it is not necessary to use rhetorical relations, syntactical rules can already realize paratactic aggregation.

An investigation into paratactic aggregation gives information about where to use this form of syntactical clause aggregation in a generated answer sentence.

Paratactic aggregation

The aggregated constituents in a paratactic construction are, as mentioned before, of equal syntactic status. In "John at an apple and an orange" for example, "an apple" and "an orange" are of equal syntactic status, because they are both labelled with the same categorial label "NP". This example, contains a coordinating conjunction of NP's and is the main operator of parataxis. Coordination is a common linguistic construction in language [Shaw, 2002] and realized by words like 'and', 'or', 'but', etc ('en', 'of', 'maar' in Dutch), the combined elements are called the conjuncts. These words combine two or more linguistic units, such as a series of clauses, phrases or words. Shaw focuses in his work mainly on the coordinator 'and' because in descriptive domains, it is the most common of the coordinators. Since the corpus used for this research is also focused on a descriptive RSI domain, the coordinator 'and' seems a good candidate for aggregation in generated answer sentences.

Coordinated conjunctions are used by humans in their everyday communication process. The coordinating conjunction is also one of the most studied linguistic phenomena as can be concluded from a literature overview in [Shaw, 2002].

To analyze the 'and' coordinating conjunction, Shaw uses two dimensions:

- With respect to syntactic properties of the conjuncts, the conjuncts are of the same basic syntactic category (conjunctions of clauses, predications, phrases or words) or the conjuncts are of different non-basic syntactic types (see the next three possibilities) and complex.
  - Non-constituent coordination: "John at *vis op maandag* en *rijst op dinsdag*" (John ate *fish on Monday* and *rice on Tuesday*).
    Over here, the conjuncts are of a complex syntactic type that consist of an "object + adverb modifier".
  - Gapping: "*John at vis* en *Bill rijst*" (*John ate fish* and *Bill rice*).
    In this complex scenario, the verb 'ate' is deleted to not repeat the same verb twice.
  - Right-node-raising: "John ving en Mary doodde de spin" (*John caught* and *Mary killed* the spider).
    Here, the identical constituent "the spider" is shared by both conjuncts.
- With respect to the intended readings of the conjoined expression, the conjuncts can be distributive or collective. In distributive coordination, the coordination of smaller parts is logically equivalent to coordination of clauses; "John en Paul vinden Mary leuk" ("John and Paul like Mary") is logically equivalent to "John vind Mary leuk" ("John likes Mary") and "Paul vind Mary leuk" ("Paul likes Mary"). However, sentences containing conjunction with collective readings cannot be analyzed as separate clauses; "Mary and Sue zijn zusters" ("Mary and Sue are sisters") is not equivalent to "Mary is een zuster" ("Mary is a sister") and "Sue is een zuster" ("Sue is a sister").

Both of these dimensions are important when creating a general conjunction algorithm and an overview can be found in [Shaw, 2002]. It shows that a general algorithm is not that easy to realize, but based on the identification of syntactic categories it is possible to realize aggregation.

3.4 Aggregation in answer presentation

As already indicated, parataxis based on the word 'and' ('en' in Dutch) looks like a promising aggregation concept to use for better readable text. However, it all depends on the retrieved text fragment if it will be necessary to apply it. This since the fragments retrieved from the internet or an encyclopedia are often in an aggregated format and don't need aggregation. This is because these fragments are realized by a human hand and therefore already in an aggregated form.

For example, sentences that contain a sequence of constituents, "Ze kunnen variëren van koude en kramp en verkleuring en stijfheid en tinteling en krachtsverlies en tot pijn of juist gevoelloosheid" will likely not occur, but will already be of the form "Ze kunnen variëren van koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid". So aggregation seems unnecessary for one sentence long answers.

When text fragments are of a different form, for example a text fragment that contains a list of two elements, the clauses can be combined with the word 'en' and form an aggregated answer instead of just concatenating all the clauses behind each other. When there exist more element in a list, it is not straightforward. It is then necessary to use the word 'and' only for the last two clauses to combine.

Example: "Wat we kunnen doen heeft vrijwel allemaal te maken met
- **het ontspannen van de spieren en geest**
- **het stimuleren van de bloedsdoorstroming.**"

In this case the text could be aggregated in the form: "Wat we kunnen doen heeft vrijwel allemaal te maken met het ontspannen van de spieren en geest *en* het stimuleren van de bloedsdoorstroming". In this form the clauses **"het ontspannen van de spieren en geest"** and **"het stimuleren van de bloedsdoorstroming"** have been identified as the parts to aggregate. Especially in a descriptive domain, such bullet lists are likely to occur, since they provide a good way of presenting an overview of different symptoms, medicines, related diseases, etc. The answers to be generated in this research will only consists of text without any markup such as bullets, tables, etc. As such it will be necessary to transform bullet lists into an aggregated text fragment, through the use of the word 'en'. Another option would be to use the bullet lists within the answer presentation, but then this markup has to be covered inside the retrieved text fragments. Of course, it would be better to keep the bullets list within a text presentation, but that is future work when other modalities for answer presentation come into view. At this stage the text fragments only contain text with no markup available and aggregation is not likely to contribute much at this stage. However, when a multi-modal presentation is being generated, it might be preferred to present the answer through a text-to-speech module. And when think of an answer that is spoken by the system, than it is necessary to cover markup within answers.

Although it depends on the format of a retrieved text fragment, it seems that syntactical aggregation through the use of the word 'en' results in more aggregated answers. But, as indicated above it is not likely to contribute to the presentation of answer texts at this stage.

3.5 Conclusion

When retrieved sentences can be aggregated in a more concise, cohesive and fluent manner, it could improve the presentation of answer texts. Describing some of the issues related to aggregation and specifically syntactical clause aggregation, indicated that different types of aggregation handle different linguistic phenomena. Syntactical clause aggregation, as described by Shaw, can be used to adapt a text fragment in a more fluent way. For this research the parataxis approach with the accompanying word 'en' is explored to link two or more linguistic units of equal syntactic status that occur in a retrieved text fragment. However, at this stage only sentence long fragments will be used for the generation of answer texts and it is therefore doubtful if it will contribute to answers being generated. When using larger text fragments and using multi-modal means for answer presentation, aggregation will be of better use.

Chapter 4

"Real-world" experiment

Based on the previous conclusions and the hypotheses in section 2.5 regarding questions and the preferred form of their answers, this experiment should confirm or disconfirm these hypotheses. The material used in this experiment consists of a set of question-answer pairs extracted from the RSI domain the experiment tests the hypotheses about Q&A matching. First the set-up of the experiment will be described and the hypotheses are outlined for clarity, then the outcome of the experiment will be presented. In the end we come up with concluding remarks regarding this experiment but also about the preferred format for presenting an answer.

4.1 Hypotheses

Topic-comment linking It is the users preference to link the question to the fragment by repeating the 'question topic' in the generated answer. By anchoring a generated answer to the question asked, the answer is placed in a clear context. By showing only the contributing information, the answer has more chance of being misinterpreted.

Topic-comment order The topic-comment pattern for the presentation of answer texts is the users preference. The less informative elements occur before the elements with an high informative value. This is closely related to the above hypothesis. If people confirm the above, it is the hypothesis that the preferred ordering of information is the topic-comment ordering: first the question topic, then the new contributing information. This to confirm that information packaging theory that counts for linguistics in general also counts specific for question answering.

Explicit use of 'yes/no' The explicit use of 'yes/no' in a congruent answer is the users preference for answering "yes/no"-questions. In other words, people don't want to conclude for themselves if a 'yes/no' can be concluded out of a presented answer sentence. It is the users preference that the system shows it explicitly.

Formulations The words used in the question must also be the used words in the answer. So words used inside the generated answer that don't match the used words in the question are not preferred. For example, the use of "veroorzaken" ("cause") instead of "leiden tot" ("give rise to") through lexical variation, or the use of "ontstaat" ("arises") instead of "kan ontstaan" ("could arise") through syntactic variation.

4.2 Material

The material that is available in the build corpus, see section 2.2.3, is being used to perform the experiment. The questions in the corpus are all accompanied with a text fragment that contains information and can be transformed in different ordering to test the hypotheses above. However, what kind of ordering are presented in the experiment is explained in the next section. Since all the material is extracted from internet, it can be stated that the contents of these text fragments can be seen as correct. So the participants in this experiment only have to mark their preferred answer and don't have to check the correctness of the contents within a generated answer. However, when it would be an evaluation of a completed QA-system, it would certainly be a good testing feature.

4.3 Experiment set-up

In the experiment, the participants can indicate for each one of the 22 questions what they see as the most preferred answer. Besides that, it was possible to add global remarks about the presented material. The answers (based on the text fragments) accompanied with each question, were transformed in different orderings and each question & answer pair contributed to the confirmation or disconfirmation of one of the four hypothesis. See the appendix for the experiment A itself and see table 4.1 for an overview of which question & answer pair is connected to which hypothesis. The experiment has been presented to the participants by mail, because at the time, it was the fastest way to get in touch with a large group of participants. There was no specific time limit to complete the test, but the estimated time to complete the test was around 20 minutes.

To evaluate	**Where evaluated**
Topic-comment linking	1,3,4,6,7,8,9,11,12,14,15,16,17
Topic-comment order	1,3,4,6,7,8,9,11,12,14,15,16,17
Yes/no explicitly	2,5,10,13,18
Formulations	19,20,21,22

Table 4.1: Hypotheses and number of questions

As can be seen in table 4.1, the topic-comment order is tested thoroughly, because it largely decides the format for the presentation of answer text. The other two hypothesis related to the explicit use of yes/no and related to the word choice within the answer, are tested less thoroughly, because they are of less influence on the syntactic pattern to present an answer. However, they are possibly important for the users preference of presenting answers.

Aggregation regarding 'syntactical clause aggregation' is NOT tested in this experiment. This because otherwise the test would become to time consuming for participants to complete and would draw too much attention away from the main hypothesis to be tested here.

I have chosen for this experiment format to test the hypotheses in the best direct way. Another possibility would have been to let the users construct an answer based on a retrieved fragment by themselves. However, that would ask a lot from the people participating in this experiment and it would also be difficult to specifically test the hypotheses. By presenting the possible answer formats, people can immediately mark the answer they prefer and it is possible to confirm of disconfirm a tested hypothesis in the end.

Participants

For the experiment, about 30 people were invited to participate and 24 people were able to complete the experiment before the two week deadline. The group consists of a variety of people with different age, different education, etc to come up with a general answer preference.

Information	Specifying	Number		Information	Specifying	Number
Gender	Female	10		**Education**	MBO-	7
	Male	14			HBO+	17
Age	< 20	1		**Region**	Twente	11
	20 - 30	14			Noord-Holland	8
	30+	9			Other	5

Table 4.2: Participants information

Participants were invited in the experiment to supply this information for further evaluation. However, it remains to be seen if things like education or age have influence on the users preference regarding answer presentation. We come back to this later on in this chapter and the provided information about the participants can be found in table 4.2.

4.4 Results

The outcome of the experiment can be found in table 4.3 and there is one thing that immediately asks for attention, without looking to the hypotheses that are being tested. *A lot of participants choose the same answer as the most preferred answer*. So the majority of the participants chose the same answer as the preferred answer. This equality in preferred answer (the difference between this preferred answer and another preferred answer is at least a factor two) counts for 73% of the questions (16 out of 22). For these questions, the number is illustrated in bold face and the % of participants that choose this answer can be found in the last column. This is interesting because it indicates that one particular answer format can satisfy a large group of people.

Q	No link	T-C	C-T	Exp y/n	No Explicit	Exact Form	No Exact	%
1)	0	**20**	4	–	–	–	–	83%
2)	–	–	–	**18**	6	–	–	75%
3)	5	**17**	2	–	–	–	–	71%
4)	1	**19**	4	–	–	–	–	79%
5)	–	–	–	**15**	9	–	–	63%
6)	5	**14**	5	–	–	–	–	58%
7)	3	**18**	3	–	–	–	–	75%
8)	1	**17**	6	–	–	–	–	71%
9)	5	**18**	1	–	–	–	–	75%
10)	–	–	–	**17**	7	–	–	71%
11)	6	**18**	0	–	–	–	–	75%
12)	2	11	11	–	–	–	–	–
13)	–	–	–	**17**	7	–	–	71%
14)	5	**16**	3	–	–	–	–	67%
15)	5	**19**	0	–	–	–	–	79%
16)	6	**15**	3	–	–	–	–	63%
17)	4	**20**	0	–	–	–	–	83%
18)	–	–	–	14	10	–	–	–
19)	–	–	–	–	–	13	11	–
20)	–	–	–	–	–	**24**	0	100%
21)	–	–	–	–	–	11	13	–
22)	–	–	–	–	–	14	10	–

Table 4.3: Experiment results

Q	= Indicates witch one of the 22 questions.
No Link	= No repetition of the question topic in the answer.
T-C	= First topic, then comment in the answer.
C-T	= First comment, then topic in the answer.
Explicit y/n	= The explicit use of yes/no in the answer.
No Explicit	= No explicit use of yes/no in the answer.
Exact form	= Using the word choice, that corresponds with the question.
No Exact	= Using a word choice that doesn't correspond with the question.
%	= % of people that have chosen the **indicated** format.

Worth mentioning is that almost all persons who completed the experiment were consistent in choosing their preference. For example, the people choosing for an answer format without 'topic-comment linking', choose this structure through the whole evaluation. A lot of people also noticed this themselves, since some remarks were written down regarding this phenomenon.

4.5 Discussion

When looking at the tested hypotheses, there are some interesting issues that come forward. For each of the hypotheses as mentioned in section 4.3, they are confirmed or disconfirmed and where necessary an explanation is provided.

Topic-comment linking This linking hypothesis has been tested in 13 questions and, in all these 13 questions, the most preferred answer included linking to the question. There were also a few participants that preferred a direct answer format consistently throughout the whole experiment. In the remarks, this group mentioned that repeating the questions is not always preferred. Certainly not, when thinking about a spoken (in terms of a dialog act) question&answer pair. Constantly repeating the question might then become irritating. However, since this research is about written text and the bigger part of the participants preferred linking to the question, it can be stated that a large group of people will prefer linking above direct answering. Exactly where this linking should occur is an other issue and has also been tested in the experiment. So the topic-comment linking hypothesis can be confirmed, based on this experiment.

Topic-comment order Testing this hypothesis is closely related to the above, as mentioned before. By using the same 13 questions, it was possible to let participants choose between 'linking' / 'no linking' and between the 'comment-topic' / 'topic-comment' pattern. When looking at table 4.3, the numbers in **boldface** indicate the (according to a group of 24 people) most preferred answer formats. 12 out of 13 of the questions that test this ordering of the topic and comment, have the "topic-comment" pattern alternative in boldface. This confirms the hypothesis the topic-comment order is preferred as the pattern for answer presentation. In the case of one question (question number 12) the "topic-comment" and "comment-topic" alternatives came both forward as the most preferred answer (both an equal score of 11). Exactly why these two alternatives are equally chosen is unclear, but probably the used text fragment is responsible for this outcome.

- **Q:** "Hoe voorkom je RSI?"
- **A:** "[Voor het voorkomen van RSI]$_{topic}$ is geen standaardoplossing aan te dragen."
- **A:** "Er is geen standaardoplossing aan te dragen [voor het voorkomen van RSI.]$_{topic}$"

Yes/no explicitly The explicit use of 'yes/no' has been tested in 5 questions and for 4 of these questions, the explicit use of 'yes/no' is indicated as preferred. However, only a 'yes' or 'no' as provided answer won't be enough. This was not explicitly tested, but can be concluded

from remarks that participants wrote down during the experiment. They stated: "A yes/no is fine, but too short for a polite answer". So for 4 out of 5 questions the explicit use of 'yes' or 'no' is desired and based on the remarks a minimal requirement. When looking at the practical side of providing the 'yes/no' explicitly it all depends on the domain and the retrieved text fragment to be able to conclude 'yes'/'no'. A lot of information regarding the RSI domain, for example, is not that clear yet for researchers, doctors, etc. So it will be a difficult task to answer a question like "Is RSI te genezen?", purely based on a retrieved text fragment, when there is not already an explicit use of 'yes'/'no' in that text fragment. All in all the hypothesis about explicit 'yes/no' use is definitely confirmed, but hard to realize in practice.

Formulations Reviewing the outcome of the questions that test hypothesis about the used formulations, reveals that it is hard to conclude a general preference. The hypothesis was that the word choice in the answer must coincide with the choice of words used in the question. However, the outcome of this experiment shows that there is no general consensus (between the group of 24 participants) about which lexical and syntactical variations are preferred. Therefore lexical and syntactical variation seems to have no big influence on the preference of the presentation of an answer text. The hypothesis about formulations can therefore be disconfirmed.

4.6 Conclusion

This experiment tells us that three out of four hypotheses, as described in section 2.5, can be confirmed and the features that are covered by these hypotheses definitely should be taken into account for the presentation of answer texts. Direct answering is an option, but is not preferable in an Q&A system when working only with written text. The structure as described by information packaging: *The less informative important elements (topic, theme) occur before the elements with a more important informative value (comment, rheme)* must be used when generating an answer to satisfy a large group of people. This structure should also form the basis for 'yes/no' questions together with an extra addition: the explicit use of 'yes'/'no' should be used whenever possible. It will depend on the knowledge of the Q&A-system and the retrieved text fragment if this is actually possible. During the surface realization of the answer generation process, lexical and syntactic variation don't seem to have a huge effect on the preferred answer.

Overall this experiment contributed to this research as a test for everything that has been encountered and described up till now regarding Q&A matching features. The experiment points out that there exists a general consensus regarding most of the hypotheses tested, but also indicates that it is impossible to satisfy everyone within a large group of people.

Chapter 5

Use of dependency trees

5.1 Introduction

Dependency as a term can be quite clear, *two objects are related to each other*, but dependency as used in linguistics cannot be described that simple. In this chapter an overview of dependency theory in the linguistic field will we given to come up with a formal definition for the term *dependency*. Since the dependency structures produced by the Alpino parser (see section 5.3) will be the structures used in this research, what follows is an explanation of the main concepts of its grammar and a deeper explanation of dependency structures as delivered by this parser. To be able to identify the patterns, relations and features, as mentioned in Chapter 2 Questions & Answer matching, the information inside the dependency structures has to be explored and the practical use of the structures needs to be discussed for use in the demonstrator to implement.

5.2 History

Dependency, although less known among linguistics than constituent analysis, is an intuitive concept. In constituency, a sentence *consists* of certain elements which in turn consist of other elements or words. In dependency, one word form *depends* on the other. Especially morphological dependencies are self-evident (a masculine article evidently depends on a masculine noun). In other words, dependency can be seen as a grammar in which individual words both act as terminal nodes and as non-terminal nodes. Terminal because they directly access the lexicon (dependency in its purest form only knows words) and non-terminal because they "require", they "subcategorize for" other words, so called dependents. The following citations in [Schneider, 1998] provide a clear perspective of where to place "dependency analysis".

> "Dependency analysis" is an ancient grammatical tradition which can be traced back in Europe at least as far as the Modistic grammarians of the Middle Ages, and which makes use of notions such as government and modification. In America the Bloomfieldian tradition (which in this respect includes the Chomskyan tradition), assumed constituency analysis to the virtual exclusion of dependency analysis, but this tradition was preserved in Europe, particularly in Eastern Europe, to the extent of grammar teaching in schools. However, there has been very little theoretical development of dependency analysis, in contrast with the enormous amount of formal, theoretical, and descriptive work on constituent structure. [Hudson, 1996]

Despite this very old tradition, going back to the Middle Ages, dependency seems to have been overshadowed by constituency more recently. Especially since the start of "modern" grammar theory towards the 20th century.

> Phrase structure representation in syntax was strongly promoted by the Structuralist school during the thirties, forties and fifties (...). It became the only syntactic representation ever seriously discussed in the work of Noam Chomsky and the Transformational-Generative School he founded in the late fifties. As a result of the triumphal offensive of the transformational-generative approach throughout the world, phrase-structure syntax forced dependency syntax into relative obscurity. [Melčuk, 1988]

These citations clearly indicate that dependency theory was already known but there was no real theoretical development in contrast to the work on constituent structure. However, a real first approach for "modern"-dependency was set up by Tesnière and later on followed by other dependency-based theories.

5.2.1 The foundation

The father of "modern"-dependency, as stated by [Melčuk, 1988, Engel, 1996], is Lucien Tesnière [Tesnière, 1959], Professor for Comparative Linguistics at the University of Montpellier from 1937 to his death in 1954. Tesnière makes the distinction between the outer form, the 'ordre linéaire', the surface string of words in the text, the 'chaîne parlée' in linear order, and the inner form, 'ordre structurale', which contains a net of relations conveying the grammatical relations in the sentence on an abstract level independent of the linear precedence in the surface text. He stresses that the domain of syntax is to describe the inner form, i.e. the structural order, while he wants to delegate the outer form, i.e. word order to morphology and phonology. In dependency theory, word order plays no primary role (it may help as a secondary role to disambiguate on the outer form if necessary), but it is not conserved in the inner form. For languages with freer word-orders than English (such as Czech, German or Dutch for example [Hajicova et al., 1998]), such a suggestion seems promising. For a thorough explanation of Tesnière approach, see [Tesnière, 1959] or see [Schneider, 1998] for a description of the basic concepts that are used in this approach.

5.2.2 Formal definition

As indicated by the above we can informally and roughly stated describe "dependency" as; *a dependency-based perspective that is realized by distinguishing a head/dependent asymmetry and describing the relations between a head and its dependents in terms of semantically motivated dependency relations.*

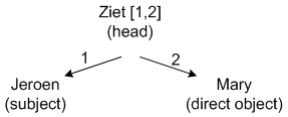

Figure 5.1: *"Jeroen ziet Mary"*

This can be illustrated by the simple sentence "Jeroen ziet Mary" ("Jeroen sees Mary") in figure 5.1. The head 'ziet' ('sees') is placed above its dependents (the subject 'Jeroen' and direct object 'Mary') and the numbers in square brackets refer to the number of dependents in a logical representation.

All the following dependency-based theories, (*Tesnière* [Tesnière, 1959], *an early formalisation* [Hays, 1964], *Meaning-Text Theory* [Melčuk, 1988], *Tree Adjoining Grammars* [Rambow and Joshi, 1994], *English word grammar* [Hudson, 1991]) describe the structure of a sentence in terms of binary head-modifier (also called dependency) relations on the words of the

sentence. All these theories gave an impulse to the dependency theory again and as such it resulted in a clear concise and useful term "dependency":

A dependency relation is an asymmetric relation between a word called head (governor, parent), and a word called modifier (dependent, daughter). A word in the sentence can play the role of the head in several dependency relations, i.e. it can have several modifiers; but each word can play the role of the modifier exactly once. One special word does not play the role of the modifier in any relation, and it is named the root. The set of the dependency relations, that can be defined on a sentence, form a tree and these are called the dependency structures.

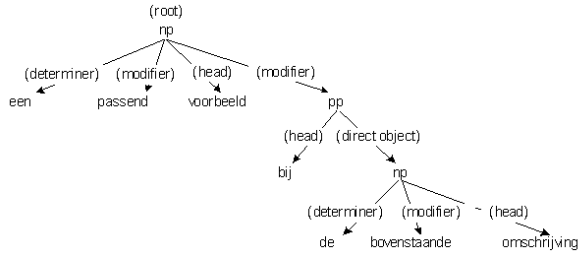

Figure 5.2: *"Een passend voorbeeld bij de bovenstaande omschrijving."*

The example structure in figure 5.2, with labels extracted from Alpino, illustrates the idea behind the definition. The Alpino-parser is a wide-coverage computational analyzer of Dutch and developed at the University of Groningen, one of the partners within the IMIX project as described in chapter 1 that also investigated the use of dependency trees for question answering. Since the Alpino-parser is the only available parser for Dutch to deliver these dependency structures and it is also used by one of the other partners in the IMIX project, it is the tool used in this research to retrieve these structures. Another parser capable of delivering dependency trees for Dutch is not known at the time of this writing. Because the Alpino-parser provides the necessary information, an explanation of the parser techniques, possible relations, possible structures, etc will follow in the next section. It is mainly based on citations out of papers that describes Alpino and as such provide a clear overview of what the parser is capable of.

5.3 Alpino

"The goal of the Alpino project is to build a wide-coverage computational analyzer of Dutch which aims at accurate, full, parsing of unrestricted text" [Bouma et al., 2001]. It produces dependency structures, thus providing an abstract and theory-neutral level of linguistic representation. The aim of Alpino is to provide computational analysis of Dutch with coverage and accuracy comparable to state-of-the-art parsers for English.

For evaluation of the coverage and a disambiguation component of the system, a testbench of syntactically annotated material is crucial. Therefore, they have annotated corpora with dependency structures since it has been suggested that dependency relations provide a convenient level of representation for evaluation of computational grammars [Carroll et al., 1998]. It also presents a neutral representation for further processing, because different annotation schemes can be chosen. The chosen annotation format is taken from the project Corpus Gesproken Nederlands (CGN, Corpus of Spoken Dutch) with some differences which are indicated in [v/d Beek et al., 2002a]. The formal syntactic annotation itself can be found in section 5.3.3.

Since the Alpino analyzer provides the dependency structures necessary for this research, an explanation of the grammar (based on HPSG), the guidelines from CGN and a deeper explanation of the dependency structures based on the formal CGN annotation will follow in the next sections.

5.3.1 The grammar

The Alpino grammar is an extension of a lexicalized grammar in the tradition of Headdriven Phrase Structure Grammar (HPSG, [Cooper, 1996] sketches the basic formalism). It is a linguistic theory that tries to formally describe and to give explanations of linguistical phenomena. HPSG uses attribute-value matrices (AVMs) inside the formalism of the grammar and in cooperation with a lexicon, it connects words to an AVM that represents the lexical properties for a specific word [v/d Beek et al., 2002b].

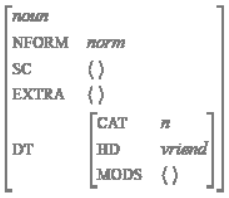

Figure 5.3: *AVM for the word 'vrienden'*

The example in figure 5.3, shows a possible AVM for the word "vrienden" ("friends"). The word is of type *noun* and this results in the attribute `NFORM`. The attribute DT represents the dependency tree within the constituent in which this word functions as the head and is responsible for realizing the dependency trees in the end.

"The Alpino grammar consists of hand-written, linguistically motivated rules and lexical types. Its formalism is carefully designed in such a way to allow linguistically sophisticated analysis as well as efficient and robust processing. The grammar contains over 100 rules (and that number is still growing as it is work-in-progress" [Bouma et al., 2001]). The grammar covers the basic constructions of Dutch as well as a wide variety of more distinctive constructions.

5.3.2 CGN (Corpus of Spoken Dutch)

The Corpus of Spoken Dutch is an annotated corpus of about one thousand hours of continuous speech, which amounts to 10 million words. The corpus is intended as a major resource, both for linguistic research and for language and speech technology. The project has added various levels of annotation to the primary speech data and the complete corpus has been orthographically transcribed; all words received a contextually disambiguated part-of-speech (POS) tag. In addition, broad phonetic transcription and syntactic annotation are provided for a representative selection of 10 percent of the data.

For syntactic annotation [v/d Wouden et al., 2002, Moortgat et al., 2002] there is a close connection with existing standards. This is not only important for recognizability, but also for accessability for users of the CGN corpus. The EAGLES-recommendations [on Language Engineering Standards, 1996] have been taken into account for realization and it is closely related to the ANS (Algemene Nederlandse Spraakkunst) [Haeseryn et al., 1997] definitions for sentence analysis. These syntactic annotation guidelines form the foundation for the dependency structures that the Alpino parser is able to produce. So, from this point forward, when speaking of Alpino dependency structures, the structures based on CGN are indicated.

5.3.3 Formal annotation

Formally a Alpino-dependency structure [Moortgat et al., 2002] is a labelled, directed, acyclic graph (DAG) and there exist disjoint collections CAT and DEP to label the nodes and the branches inside these structures.

- Nodes: CAT = POSCAT $\bigcup$ PHCAT: categorical labels (c-labels) are the conjunction of lexical (part-of-speech) and phrasal labels.
- Branches: DEP: dependency labels (d-labels).

There exist junctional ("gelede") and unjunctional ("ongelede") dependency structures. An unjunctional dependency structure is a node with a c-label from POSCAT. In other words, a subgraph that only contains a word. The elementary building stones of the junctional dependency structures are called the local dependency-domains. The mother nodes of a dependency-domain are labelled with a phrasal label from PHCAT. The daughters have c-labels from CAT. The d-labels of the mother-daughter branches are formed through a head, together with the complements and the modifiers of that head. The complements and modifiers form the relations that phrases fulfill in relation to the head.

- Head - The head of a dependency-domain projects the c-label of the mother node.
- Complements - The complements pattern determines the interpretation of the head in terms of dependency domains. In general complements refer to essential participants in a situation described by the verb.
- Modifiers - Modifying elements don't modify the c-label of a mother node; they can (in principle) be ignored without any effects on the dependency-domain. As a result of this, the same modifiable label can occur more then once within a domain. They refine the description of the situation.

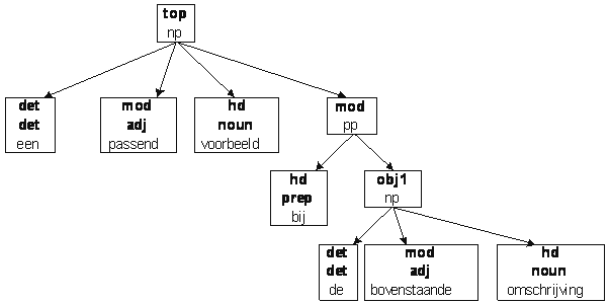

Figure 5.4: *"Een passend voorbeeld bij de bovenstaande omschrijving."*

By identifying the labels in the example in figure 5.4, it becomes clearer what terms are possible and where they can be found inside a dependency structure as delivered by Alpino. As can be seen, Alpino delivers structures in a format that might not look directly a structure with dependency relations. However, the dependency relations are placed inside the leafs of the structure for a better presentation format. When comparing figure 5.2 (page 38) and figure 5.4 (page 40) it becomes clear how the relations are illustrated and that Alpino is capable of adding Part-Of-Speech tags to words in the dependency structure. Table 5.1 provides an overview of the terms and tags that can be identified in this example. The table only shows a small part of the possible labels & tags, but a complete list, accompanied with clear examples for each of these possible c-labels & d-labels, can be found in [v/d Wouden et al., 2002] and in appendix B.

A critical notion is that the annotation provides a dependency structure and not a constituents or functional structure (not a tree structure or constituent structure as seen from a classic perspective as indicated by [Tesnière, 1959]). Branches in the tree can cross each other and daughters can have more than one parent. The terms branches, daughters and parents can be explained by the example in figure 5.5. A branch is for example the line between [VC, inf] and [HD, verb, kom_voor], the

Term	Element
CAT	POSCAT ⋃ PHCAT
POSCAT	det(erminer), adj(ective), noun, prep(osition)
PHCAT	np (noun phrase), pp (preposition phrase)
DEP	det(erminer), mod(ifier), hd (head), obj1(direct object)

Table 5.1: Terms & elements identified in the example

rectangle [HD, verb, kom_voor] represents a daughter of the parent [VC,inf]. The phrase labeled with the *su*-relation in this example, "kruisende takken" has the 'top' as its parent, but it also forms the 'subject' within the 'vc'. This is indicated by the **1** in the example. By indicating this second relation for "kruisende takken" by the number **1**, the structures will not be disturbed by crossing branches. Another interesting point, in this example, is the use of uninflected word forms (the use of "kruis" instead of "kruisende" for example). In the XML format, both the 'exact word' and stem are stored to be able to do analysis. In this XML format, the word order is made explicit in such a way that surface patterns can be easily extracted. The XML format has the focus in the next section.

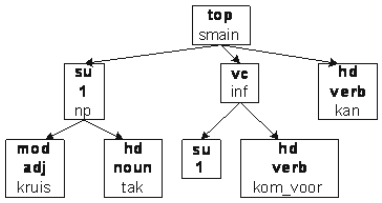

Figure 5.5: *"Kruisende takken kunnen voorkomen."*

5.3.4 Dependency output

Dependency structures make explicit the dependency relations between constituents in a sentence. Each non-terminal node in the dependency structure consists of a head-daughter and a list of non-head daughters, whose dependency relation to the head is marked.

Addition of DT attribute

To be able to produce the dependency structures with the Alpino grammar, a new level was added to the original Alpino parser [Bouma et al., 2001]. An attribute DT dominates a dependency structure, with attributes of the lexical head (HD) and the various dependents. The value of a dependent attribute can be a dependency structure or a leaf node consisting of a POS-tag and word only.

The construction of the dependency structures is driven by the lexicon of Alpino. For each subcategorization type recognized in the lexical hierarchy a mapping between elements on the list-valued feature, which specifies basic subcategorization properties (SUBCAT) and attributes of DT, is defined. In some cases, the addition of dependency structures leads to more fine-grained distinctions. For instance, PP-arguments can be divided into two classes; PC (prepositional complement) or LD (locative or directional complement), where the distinction between these two is primarily semantic in nature. So the addition of the DT attribute is necessary for Alpino to construct a dependency structure:

1. The first step consists of lexical analysis.
2. In the second step a parse forest is constructed.
3. The third step consists of the selection of the best parse from all the parsed forests.

XML format

The dependency structures generated by Alpino are in the end stored as XML documents. A suitable query language for these XML documents is the XPath standard, which can be used to formulate queries over a large collection of XML documents [Bouma and Kloosterman, 2002]. However, for clarity and reusability, they transformed the data into a compact XML representation that is suited for linguistic exploration. The new designed DTD[1] is suited for efficient querying, since all the information is now directly stored in clear XML structures.

This direct encoding of dependency trees in XML documents makes information about word order explicit, which allows queries concerning dependency, linear order, and discontinuous constituency to be stated. This direct encoding can best be illustrated by the example sentence *"Dit in verband met de gemiddeld langere levensduur van de vrouw"* in figure 5.6.

```
<top>
  <node rel="top" cat="du" begin="0" end="11">
    <node rel="dp" pos="noun" begin="0" end="1" root="dit" word="Dit" />
      <node rel="dp" cat="pp" begin="1" end="11">
        <node rel="hd" pos="prep" begin="1" end="4" root="in verband met" word="in verband met" />
        <node rel="obj1" cat="np" begin="4" end="11">
          <node rel="det" pos="det" begin="4" end="5" root="de" word="de" />
          <node rel="mod" cat="ap" begin="5" end="7">
            <node rel="mod" pos="adv" begin="5" end="6" root="gemiddeld" word="gemiddeld" />
            <node rel="hd" pos="adj" begin="6" end="7" root="lang" word="langere" />
          </node>
          <node rel="hd" pos="noun" begin="7" end="8" root="levensduur" word="levensduur" />
          <node rel="mod" cat="pp" begin="8" end="11">
            <node rel="hd" pos="prep" begin="8" end="9" root="van" word="van" />
            <node rel="obj1" cat="np" begin="9" end="11">
              <node rel="det" pos="det" begin="9" end="10" root="de" word="de" />
              <node rel="hd" pos="noun" begin="10" end="11" root="vrouw" word="vrouw" />
            </node>
          </node>
        </node>
      </node>
    </node>
  </node>
  <sentence>Dit in verband met de gemiddeld langere levensduur van de vrouw .</sentence>
  <comments>
    <comment>Q#65|Dit in verband met de gemiddeld langere levensduur van de vrouw .
             |1|1|-0.4588297573000001|1.1250930358000004</comment>
  </comments>
</top>
```

Figure 5.6: XML format

As can be seen in the above figure, all the words are covered in leafs delimited by `<node ... />` while parents are delimited by `<node ...>` and `</node>`. Therefore all the information covered between `<node \dots >` and `</node>` can be easily recognized as elements that belong to a certain relation. So when daughters and parents can be identified, it is possible to create a tree structure once again in a straightforward way. Besides, all the necessary information can be found as attributes in the XML structure.

[1] **D**ocument **T**ype **D**efinition, the legal building blocks of an XML document; it defines the document structure with a list of legal elements.

5.4 Practical use of Alpino structures

In chapter 2, an overview has been given regarding questions and answers and the features that play a role between congruent question-answer pairs. The hypotheses that followed out of this overview have been tested by a short experiment and form the foundation for a practical implementation. However, to be able to computationally generate answers based on confirmed hypotheses, the necessary features and properties have to be identified by using dependency structures as provided by Alpino.

As mentioned in the previous section, a dependency structure contains the original sentence, the relations between the phrases and words in the sentence, POS-tags for all the words, the exact used word and the stem for all words. Also the linear order, of the words in the sentence, is made explicit by the chosen DTD for the XML format. In short, it looks like all the necessary information is available in dependency structures and it should be possible to extract the necessary properties and features of questions and text fragments. The issue that still remains is how to actually use this information? What exact information needs to be identified and how to extract is out of the dependency structures, is explained in the next chapter, design & implementation.

Chapter 6

Design & Implementation

6.1 Introduction

This chapter describes the design process of the modules that have been implemented based on all of the previous analysis done regarding "Question & Answer matching", "Dependency" and the outcome of the experiment. Based on the research done regarding "Aggregation", could be concluded that the aggregation of sentence long answers doesn't contribute much. Therefore aggregation will not be applied in the demonstrator. The demonstrator is focused as first attempt on the realization of an answer that is based on the topic-comment pattern. So the issues related to grammatical correctness and explicit use of 'yes/no' for "yes/no"-questions is not implemented in the demo. Although these issues are the users preference, see the experiment in chapter 4, the topic-comment has the most influence on the generation of an answer. Besides, implementing all these issues in the demonstrator would cost to much time to complete within a reasonable amount of time. First an overview of the different modules, necessary for a kind of NLG pipeline, will be presented and this overview also points out what module is responsible for which tasks to generate congruent questions & answer pair. The main modules in this overview are divided into an *Analyzer* and into a *Formulator* module. The next section will specifically describe the algorithm to revise a selected fragment into a natural answer for the selected question, followed by the used programming language (Java in this case) and the necessary packages for XML processing (JDOM).

6.2 Modules overview

Working with separate modules seems a good approach especially when looking at other attempts [Wilcock, 2001] to realize a pipeline that is able to do efficient processing with XML. In the case of realizing a natural text answer for a question, two main modules can be identified:

Analyzer An analyzer that analyzes the question and also provides the necessary properties of the retrieved text fragment. What exact properties will be specified in section 6.2.3.

Formulator The formulator is capable of generating the answer and uses the properties collected by the analyzer module. The formulator module contains the main revision algorithm that is responsible for the identification process and for applying transformations. A description of the algorithm & transformations that are identified and applied can be found in section 6.3.

6.2.1 Information flow

The prototype works with the questions and text fragments from the corpus (as described in section 2.2.3) so all the possible questions and relevant fragments have already been annotated by Alpino. These questions and fragments are stored as XML files and all satisfy the DTD (Document Type Definition) in figure 6.1.

```
<!ELEMENT top (node+, sentence, comments)>
<!ELEMENT node (node*)>
<!ATTLIST node
    rel       CDATA   #REQUIRED
    index     CDATA   #IMPLIED
    pos       CDATA   #IMPLIED
    cat       CDATA   #IMPLIED
    begin     CDATA   #IMPLIED
    end       CDATA   #IMPLIED
    root      CDATA   #IMPLIED
    word      CDATA   #IMPLIED>
<!ELEMENT sentence (#PCDATA)>
<!ELEMENT comments (comment*)>
<!ELEMENT comment (#PCDATA)>
```

Figure 6.1: DTD of Alpino annotation

How the question and fragment will eventually result in a presented answer, can be seen in the overview in figure 6.2. First a question and a fragment have to be selected manually to mirror a possible output from the retrieval phase of the Q&A system. When the selection is finished, a separated analysis of the question and the fragment will directly be initiated by the *Analyzer*. The properties extracted from this analysis will be presented in the demonstrator to be able to do error-checking and to improve the way the revision algorithm works. Especially since the best approach to create a robust algorithm is to start with a "simple" question and a "simple" fragment and to extend this as the implementation phase passes on. When all the analysis data is presented, the generation pipeline jumps to the next main module, the *Formulator*.

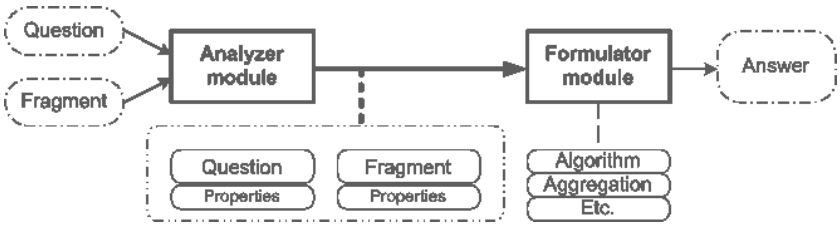

Figure 6.2: Modules overview

Within the formulator module, the main revision algorithm is activated and the important transformations are identified and applied. The output of the algorithm is the answer to present as a congruent answer for the original question. The format, of this output, will most logically be XML as the whole pipeline will be in XML. By choosing for XML as output, further processing by other modules within IMOGEN should not be a problem. An interesting aspect for these modules is to extend them with smaller parts that are each responsible for a specific task. Then you can think of parts that are responsible for aggregation, explicit use of 'yes/no', etc. This is especially interesting for future development, because for now the prototype is focussing on revision of the fragment structure based on syntactic patterns (revision algorithm). This idea of extending is indicated by the boxes under the formulator module.

6.2.2 Used definitions

Before further information is given regarding the *Analyzer* and the *Formulator* it is time to look back at the definitions regarding the "topic-comment" structure and how these are interpreted and used in the implemented demonstrator:

- **Topic** The topic provides the anchoring role to the previous discourse and usually contains old, already known information.
- **Comment** The comment is what expresses the new contributing information about the topic.
- **Question Topic** The thing someone intends to increase his/her knowledge about by using the question.
- **Answer Topic** A repetition of the question topic into the generated answer to realize the anchoring role to the question.
- **Answer Comment** The information in the generated answer out of the text fragment, that expresses the actual information asked about in the question.

With these definition in mind is it possible to define the actions the analyzer and formulator are responsible for.

6.2.3 Analyzer

As mentioned before, the analyzer is responsible for identifying the questions and text fragment properties that are necessary for the formulator to generate the final answer. These question and text fragment properties have been described in chapter 2 and a summary of the matching features (based on these properties) was presented in section 2.4. First, a short description of the properties that are identified and in section 6.3.1 an explanation is provided of how to determine these properties in practice.

Question type An identification of the possible three questions types "yes/no","choice" and "wh-word" questions.

Answer type As explained before, there exists an expected answer type for each question. By identifying this type it is possible to look up the suggested answer pattern. This is straight forward for boolean and alternatives answer types, but also possible for wh-word questions.

Question topic Identification of the topic of the question, to repeat this topic in the to generate answer based on the topic-comment pattern. Including the topic of the question in the generated answer.

Answer pattern All the answer patterns should satisfy the topic-comment order. The generated answer should ideally be an ordered sentence that starts with the *answer topic* from the question followed by the *answer comment* that actually contains the information that is asked about.

Fragment type For future use it is preferable to choose between fragments with the length of text snippets (the fact that is asked about), sentences (the sentence that contains the information that is asked about) and paragraphs (the paragraph that contains the relevant information) fragments. That's why the identification of the fragment type is. However for this proof-of-concept, sentence long fragments are used and no further time is invested in fragment type identification.

Fragment subjects A lot of retrieved fragments probably contain elements that also occur in the question topic, because that is why a specific fragment is retrieved in the first place (based on basic keyword comparison). By identifying the subjects of a fragment (the identification of these subjects doesn't say anything about an identified topic without further analysis, because Alpino can assign multiple *su*-relations within a sentence.), it is possible to compare these fragment subjects with the question topic elements. This comparison should provide information about the probability that an identified subject indeed can correspond to the question topic. When this is known, it is also possible to identify the information that was originally asked for (the 'comment') and an answer based on the topic-comment pattern can be generated.

6.2.4 Formulator

The formulator module is responsible for activating the main revision algorithm in the next section. The formulator receives all the necessary properties from the analysis phase, that provides the properties of the question and of the text fragment. Based on these properties the revision algorithm identifies the elements in the dependency tree that are candidates for transformations (like add, move, del, etc) in order to produce the answer based on the "topic-comment" structure.

After this identification phase, the transformations are applied and walking through the dependency tree results in adaptations in the tree. By walking through the tree one last time, after all transformations have been performed, the new generated tree is stored in the XML format and is again suitable for future processing. How all of this is handled can be found in the next section.

This process can be looked at in a systematic way and that results in the following global steps to be taken:

Step 1: Collect properties from the Analyzer module.

Step 2: Compare fragment with the suggested matching features.

Step 3: *IF* Fragment already fulfills the matching aspects goto step 5.

ELSE Identify the necessary transformations.

Step 4: Realize transformations by changing the fragment structure.

Step 5: Output XML file with the best possible congruent answer.

6.3 Revision algorithm

The revision algorithm has the main focus in the demonstrator and is capable of handling revision within the dependency tree of the selected answer fragment. This algorithm needs the collected information during the analysis stage and uses that information together with Q&A matching features to generate an answer. This section will explain first how the properties are identified during the analysis phase and secondly will explain the revision algorithm that makes use of this information. A description of the *relations* used by Alpino that are identified in this section, can be found in appendix B.

6.3.1 Retrieved properties

Each property, as mentioned in section 6.2.3, has to be retrieved purely based on the identification of words and syntactic patterns that can make a distinction between the different instances (for example: 'what question type', 'what answer type', etc.). The following overview presents that

information and since the properties *answer pattern & fragment type* only have one possible value, they won't be further explained.

Question type The three basis question types as identified in section 2.2.2 are identified by the occurrence of specific dependency relations or specific words:

- The *whq*-relation ("constituentvraag", a sentence is identified as a wh-word question / constituent question) in the case of a "wh"-question.
- The *sv1*-relation ("zin met V op de eerste plaats", sentence with a verb on the first spot) in the case of a "yes/no"-question.
- The occurrence of the 'of' ('or') word in a sentence strongly indicates a 'choice"-question.

Answer type The closely related expected answer type follows directly from the question type in the case of "yes/no" and "choice"-questions. In the case of a "yes/no"-question the expected type is boolean and in the case of a "choice"-question it is one of the available alternatives in the question itself. The "wh"-questions need further investigation, but the expected answer type follows directly from the used 'wh-word'. An overview of these possibilities can be found in table 6.1. Although these expected answer types are identified, they are not used in the demonstrator. However, for future use it adds a possibility for semantical checking of identified subjects. If for example a subject like "hundred people" is identified as comment in de fragment and has the semantic tag `<con_amount>`, it is likely to be indeed a comment for the question "How many people ...?". However, if another tag is provided it conflicts with answer type and further analysis is necessary.

Question Word	In English	Answer Type
Wanneer	When	Time
Wie	Who	Person
Waarom	Why	Reason
Wat	What	Object/Idea/Action
Welke	Which	Alternatives
Waar	Where	Place
Waar...	Where ...	...
Hoe	How	Manner
Hoeveel	How many/much	Amount
Hoe lang	How long	Duration
Hoe vaak	How often	Frequency
Hoe ...	How ...	...

Table 6.1: Possible wh-words and suggested answer types

Question topic The thing someone intends to increase his/her knowledge about by using the question. So when this is known, the identification of the question topic can be realized by searching the dependency trees specific for *noun* & *name* POS-relations and for the *NounPhrase* category relation. By searching for these POS-relations, the the content words of the question are identified and these represent the information need that is expressed by the question. This process of topic identification results in question topics, like the questions topics presented in table 6.2 on the next page. These are question topics as identified by the implemented demonstrator.

Question	Question Topic
“Wat zijn symptomen van RSI?”	“symptomen van RSI”
“Welke factoren zijn verantwoordelijk voor RSI?”	“factoren verantwoordelijk voor RSI”
“Hoeveel mensen hebben last van RSI?”	“mensen last van RSI”
“Wat is de definitie van RSI?”	“de definitie van RSI”
“Zijn er medicijnen voor RSI?”	“medicijnen voor RSI”

Table 6.2: Question topics as identified by the demonstrator

Fragment subjects Identifying the fragment subjects, purely based on syntax can be done by identifying the *subject*-relation inside the text fragment. These subjects are investigated by the algorithm and might be identified to function as ‘topic’ or as ‘comment’. It depends on further analysis if and what status is assigned to an identified subject. One important remark, the identification of these subjects doesn’t say anything about an identified topic yet and Alpino can assign multiple *su*-relations within a sentence. However, the subjects provide a good starting point within the main revision algorithm, since subjects usually indicate what a fragment is about. By further analysis of these subjects, it is therefore possible to identify in the fragment, what can function as answer comment and what part of the text functions as question topic.

6.3.2 Algorithm

1. Making use of the identified question topic, it is possible to check all the words in the fragment sentence (that meet the constraint that the POS-tag must be noun/name/verb, the so called “content words”) with the question topic elements. These matching elements, checked through regular expressions, are marked and used in the next phase of the algorithm.

2. This step identifies what information regarding the ‘topic - comment’ structure is available inside the text fragment. It is based on scenario’s and can best be described by pseudo code:

```
IF (Q <-> A match found){
    FOR (all identified matches) {
        IF (match found inside identified fragment subject) {increase score of subject}
    }
    IF (subjects found in fragment)
        FOR (all found subjects) {
            IF (score of subject > stored possibleTopic) {store possibleTopic = current subject
                algorithmCase = "* Match found and one subject very likely to be topic *"}
            ELSE IF (score of subject < stored possibleTopic && #words of subject > 2) {
                store possibleComment = current subject
                algorithmCase = "* Match found and one subject very likely to be comment *"}
            ELSE { create newPossibleTopic for found match & store possibleTopic = newPossibleTopic
                algorithmCase = "* Match found but not inside an identified subject *"}
        }
    ELSE (match found but no subjects found in fragment) {
        create newPossibleTopic for found match & store possibleTopic = newPossibleTopic\\
        algorithmCase = "* Match found but no subject identified *";
    }
}
ELSE IF (no match found && subjects found in fragment){
    IF (1 subject found && #words of subject <= 3) { identifiedTopic = found subject
        algorithmCase = "* No match is found, 1 short subject found -> topic *"}
    ELSE IF (1 subject found && #words > 3) { identifiedComment = found subject
        algorithmCase = "* No match is found, 1 long subject found -> comment *"}
    ELSE IF (multiple subjects found) {
        FOR (all found subjects) {store identifiedTopic = shortest subject) {
            algorithmCase = "* No match is found, subjects were found -> shortest is topic *"
        }
    }
}
ELSE (total fragment identified as comment) {
    create newPossibleComment for total fragment && store identifiedComment = newPossibleComment
    algorithmCase = "* No subject and no match is found, total fragment is comment *"
}
```

Although the pseudo should be easy to read, it needs explanation. When looking at the pseudo code, there exist three global states and there exist eight case scenarios. The tags of these scenarios give information of the choices made within the algorithm. With the explanation provided below and with the provided example in this section, the working of the algorithm should be clear.

(a) The first state is only entered, when there is found a match between elements in the text fragment and elements in the question topic (Q <-> A match found).

Once in this state, all the found subjects that contain an element that matches with the question topic receive an increased score. By increasing the score it is possible to make a distinction between subjects that can function as answer comment (these will have a low score) and subjects that match with the question topic (these will have a high score).

When subjects have been identified in the text fragment it is possible to identify the subject with the highest score as match with the question topic (the scenario "* Match found and one subject very likely to be topic *") or the subject with the lowest score as answer comment (the scenario "* Match found and one subject very likely to be comment *").

As can be seen in the algorithm, there are still two possible scenarios, "* Match found but not inside an identified subject *" and "* Match found but no subject identified *". These are chosen when a match between the text fragment and the question elements has been found, but cannot be traced back to a subject within the text fragment. However, since the match is found it strongly indicates a match with the question topic and the element is marked as match.

(b) The second state is entered when no match was found between the text fragment and the question topic && there were *su*-relations identified in the text fragment.

In this state it is still possible to make a distinction between subjects that can function as answer comment and as question topic. However, this time not based on the score of a particular subject, but based on the number of words covered in the subject.

When there is one subject identified in the text fragment, it will be marked as answer topic for subject that consists of less than three words (the scenario "* No match is found, 1 short subject found $->$ topic *"). Subject that consists of more than three words are marked as answer comment (the scenario "* No match is found, 1 long subject found $->$ comment *"), because it is the assumption that long identified subjects are likely to function as answer comment instead match with the question topic. The same approach is taken when multiple subjects have been identified, but this time the shortest subject is marked as match with the question topic (the scenario "* No match is found, subjects were found $->$ shortest is topic *").

(c) The third state is entered when none of the above could be entered.

This is the case when no matches were found and there were also no *su*-relations identified within the fragment. However, the fragment has been delivered as retrieved material and as such contains relevant information. Therefore, the total fragment will be marked as 'answer comment'. This is scenario "* No subject and no match is found, total fragment is comment *".

3. What follows after this identification phase is the realization of the necessary transformations. An identifiedTopic or identifiedComment is known at this stage and there are therefore two options available:

   - The identifiedTopic is known and replaced by the identified question topic to form the answer topic. Because the identifiedTopic is known, the comment part within the fragment can also be extracted based on index values within the fragment.
   - The identifiedComment is known to form the answer comment and the identified question topic is inserted before the answer comment. Because the identifiedComment is known, the part within the fragment that functions as topic can also be extracted based on index values within the fragment.

   Based on the available indexes, it is further possible to do a kind of "cleaning phase". This means that all the elements in the XML tree that aren't contained within an identifiedComment are set to blank.

4. The last step is activated when the begin index of the topic exceeds the 4th word in the generated answer sentence. Otherwise 'topic fronting' is necessary to comply with the topic-comment order. When not performing topic fronting in this case, the generated answer would have the comment-topic order.

Explanation by examples

The best way to further explain the algorithm is probably through the use of an example. This section will walk through the algorithm and specify the actions that are performed:

Question 1: "Wat zijn verschijnselen van RSI ?"
Fragment 1: "Trillingen in handen en armen, tintelingen, pijn, stijfheid, krachteloosheid in met name de arm (muisarm) zijn mogelijke verschijnselen."

To identify the properties that are available in this question and fragment, the Alpino structures of the question and the fragment are analyzed by the Analyzer.

```
<top>
  <node rel="top" cat="whq" begin="0" end="5">
    <node rel="whd" index="1" pos="noun" begin="0" end="1" root="wat" word="Wat" />
    <node rel="body" cat="sv1" begin="0" end="5">
      <node rel="su" index="1" />
      <node rel="hd" pos="verb" begin="1" end="2" root="ben" word="zijn" />
      <node rel="predc" cat="np" begin="2" end="5">
        <node rel="hd" pos="noun" begin="2" end="3" root="verschijnsel" word="verschijnselen" />
        <node rel="mod" cat="pp" begin="3" end="5">
          <node rel="hd" pos="prep" begin="3" end="4" root="van" word="van"/>
          <node rel="obj1" pos="name(ORG)" begin="4" end="5" root="RSI" word="RSI" />
        </node>
      </node>
    </node>
  </node>
  <sentence>Wat zijn verschijnselen van RSI ?</sentence>
  <comments>
    <comment>Q#100|Wat zijn verschijnselen van RSI ?|1|1|-0.27203759439999986|1.6591170720000004</comment>
  </comments>
</top>
```

Alpino structure of the question.

Identified properties:

Question type: **Wh-word**
Based on *whq*-relation and the word "Wat" in the dependency tree.
Question topic: **'verschijnselen van RSI'**
Based on *NounPhrase*-relation in the dependency tree.
Answer type: **(Object/Idea/Action)**
Based purely on the identified wh-word 'Wat'.
Fragment subject: **'Trillingen in handen, en armen, tintelingen, pijn, stijfheid, krachteloosheid in met name de arm muisarm'**
Based on identified *su*-relation in the dependency tree.

```
<top>
  <node rel="top" cat="smain" begin="0" end="24">
    <node rel="su" cat="np" begin="0" end="20">
      <node rel="hd" pos="noun" begin="0" end="1" root="trilling" word="Trillingen" />
      <node rel="mod" cat="pp" begin="1" end="20">
        <node rel="hd" pos="prep" begin="1" end="2" root="in" word="in" />
        <node rel="obj1" cat="conj" begin="2" end="20">
          <node rel="cnj" pos="noun" begin="2" end="3" root="hand" word="handen" />
          <node rel="crd" pos="vg" begin="3" end="4" root="en" word="en" />
          <node rel="cnj" pos="adj" begin="4" end="5" root="arm" word="armen" />
          <node rel="cnj" pos="noun" begin="6" end="7" root="tinteling" word="tintelingen" />
          <node rel="cnj" pos="noun" begin="8" end="9" root="pijn" word="pijn" />
          <node rel="cnj" pos="noun" begin="10" end="11" root="stijfheid" word="stijfheid" />
          <node rel="cnj" cat="np" begin="12" end="20">
            <node rel="hd" pos="noun" begin="12" end="13" root="krachteloosheid" word="krachteloosheid" />
            <node rel="mod" cat="pp" begin="13" end="20">
              <node rel="hd" pos="prep" begin="13" end="14" root="in" word="in" />
              <node rel="obj1" cat="np" begin="14" end="20">
                <node rel="mod" pos="adv" begin="14" end="16" root="met name" word="met name" />
                <node rel="det" pos="det" begin="16" end="17" root="de" word="de" />
                <node rel="hd" pos="noun" begin="17" end="18" root="arm" word="arm" />
                <node rel="mod" pos="noun" begin="19" end="20" root="muis_arm" word="muisarm" />
              </node>
            </node>
          </node>
        </node>
      </node>
    </node>
    <node rel="hd" pos="verb" begin="21" end="22" root="ben" word="zijn" />
    <node rel="predc" cat="np" begin="22" end="24">
      <node rel="mod" pos="adj" begin="22" end="23" root="mogelijk" word="mogelijke" />
      <node rel="hd" pos="noun" begin="23" end="24" root="verschijnsel" word="verschijnselen" />
    </node>
  </node>
  <sentence>Trillingen in handen en armen , tintelingen , pijn , stijfheid , krachteloosheid
           in met name de arm ( muisarm ) zijn mogelijke verschijnselen</sentence>
  <comments>
    <comment>Q#101|Trillingen in handen en armen , tintelingen , pijn , stijfheid , krachteloosheid in met
    name de arm ( muisarm ) zijn mogelijke verschijnselen|1|1|-0.42691985000000016|0.2829893779999988</comment>
  </comments>
</top>
```

Alpino structure of the fragment.

Revision algorithm:

```
1.  Match found for ''verschijnselen''.
2.  Only one fragment subject found with no increased score.
    Scenario: * Match found and one subject very likely to be comment *.
3.  IdentifiedComment is known with index from 'Trillingen' till 'muisarm'.
        Insert ''verschijnselen van RSI'' on top.
        Start cleaning of ''zijn mogelijke verschijnselen''.
4.  No topic fronting, because topic already placed in front segment.
```

As illustrated, the first step in the algorithm, identifies one match between the fragment and the question elements $->$ the identification of "verschijnselen". However, this element with index 23

is not contained in the identified fragment subject, which starts from index 0 till index 20 (see the Alpino structure of the fragment). Therefore the score of the subject is still the original score and the algorithm identifies the subject as identifiedComment in the second step (the scenario "* Match found and one subject very likely to be comment *"). In step three it is possible to insert the question topic "verschijnselen van RSI" in front of the identifiedComment. Because the indexes of the identifiedComment are also known, from 0 till 20, it is possible to clean the indexes that are greater then 20. So in this example, the phrase "zijn mogelijke verschijnselen" is set to blank. The last step, responsible for topic fronting is not activated, because the answer topic is already added before the answer comment.

Answer 1: "[verschijnselen van RSI] trillingen in handen, en armen, tintelingen, pijn, "stijfheid, krachteloosheid in met name de arm muisarm."

As becomes clear out of this example, the extraction of the properties is purely based on the available information in the dependency tree. Hence, the more information is added in syntactical of semantical form, the better choices can be made for the identification process of a possible 'topic' or 'comment'. When this extra semantical information would be available, it is especially useful if one of the case scenario's "* No match is found, 1 short subject found $->$ topic *", "* No match is found, 1 short subject found $->$ topic *" and "* No match is found, subjects were found $->$ shortest is topic *" is selected by the algorithm. Since the choices made in these scenario's are based on the word length of subjects and that is not the perfect way to do identification. For example the addition of semantic tags (as used in the ROLAQUAD project, that also is a partner within the IMIX program), is likely to contribute to the identification choices made. By the use of tags like `<con_disease>`, `<rel_is_symptom_of>`, `<con_person>`, etc it would be possible to not only compare the syntactic information of the question topic with the fragment subjects, but to also compare the semantic tags. So by adding extra information, the identification process within the algorithm could be improved.

This is only one example that shows the algorithm for the scenario: "* Match found and one subject very likely to be comment *". For better understanding of the other possible algorithm scenario's I'd like to refer to appendix C for more examples.

6.3.3 Transformations

The processing of the transformations occurs in two different phases:

1. Phase 1: identification of the elements (nodes) in the dependency tree that need an add, move, delete, change operation. This by marking the identified topic or comment. This is handled in step 2 of the algorithm.
2. Phase 2: handling the transformations through which the content of the dependency tree will change in such a way that a correct sentence structure still exists after the transformations and that the answer fragment fulfills the matching features. This is handled by step 3 & 4 of the algorithm.

The transformations that are identified at this stage, are all related to the topic-comment order. That means, identification of the answer topic and the answer comment parts that need to be in the generated answer and ordering them. More possible transformations were already identified in the experiment (see chapter 4), but to implement them all in the demonstrator and make it robust is very time consuming and an impossible task to handle within a reasonable time period. For example the adaption of the verbs into plural or singular form, might be a very useful transformation. Or the semantical analysis of the fragment in such a way, that a explicit yes/no can be given.

6.4 Implementation

6.4.1 JDOM

To be able to realize this design, I have chosen for the programming language Java and an additional package called JDOM[1] to handle the information in the XML documents. JDOM is both Java-centric, Java-optimized and open source. It behaves like Java, it uses Java collections and it is easy in use for walking through XML documents. So, it is a pure Java API for parsing, creating, manipulating and outputting XML documents.

While JDOM integrates with existing standards such as the Simple API for XML (SAX) and the Document Object Model (DOM), it is not an enhancement to those API's but it is an alternative API[2]. Rather, it makes reading and writing XML data possible without the complex and memory-consumptive options that the other current API offerings provide (like SAX and DOM). A disadvantage might be that it has no parser, because it has to depend on a SAX parser to parse XML documents and to eventually build JDOM models from them. However, JDOM can use nearly any parser, so there is always a parser available when using the JDOM package. Also interesting is the fact that the JDOM package has a Java Specification Request (JSR-102) which opens the door for JDOM to be incorporated into the Java platform in a future release of the Java package.

Practical use of JDOM

Once an XML document has been loaded into memory, whether by creating it from scratch or by parsing it from a stream, JDOM can modify the document, because it is fully readable ad writeable. The JDOM model can be seen as an abstract data structure that represents XML documents as trees composed of nodes (elements, attributes, comments, and so forth). All these nodes of the tree can be moved, changed, deleted and added to as long as the common XML restrictions are taken into account. For instance, an attribute can be added to an element but not the other way around. During the revision stage, all these operations (moving, changing, deleting, adding) will be necessary to revise a retrieved text fragment into the final congruent answer.

When finished working with a JDOM tree, it is possible to write it back out to disk or onto a stream for further internal processing. Because processing an XML file is equal to walking through a tree, it is rather straightforward to analyze the annotated questions and answers for the specified properties. When the properties are identified, is is possible to perform transformations in the XML fragment files and to eventually provide as output an XML file with a congruent answer for the question asked. All in all, the combination of Java and JDOM should provide a steady foundation for exploring the practical possibilities, based on the previous chapters (question&answer matching, experiment, dependency).

6.5 Conclusion

At this stage the demo functions as a "proof-of-concept" in such a way that it shows that the question topic and answer comment can be identified. Through this identification process, the topic-comment order can be achieved by ordering these identified parts. However, the generation of a correct answer will most of the times not be satisfying, since grammatical correctness is just as important. These issues will be handled in chapter 7, but for sure it can be stated that a lot of work still needs to be done in realizing a natural presentation of answer text. However, the identification and generation of the topic-comment pattern provides a foundation for future work.

[1] See http://www.jdom.org/ for specific information and the installation package.
[2] See "Processing XML with Java" [Harold, 2001] for more information about SAX and DOM.

Chapter 7

Assessment

7.1 Introduction

This assessment has as function to evaluate the coverage of the algorithm, to evaluate the correctness of the available scenario's and to pinpoint the issues that need to be addressed in future work. It investigates the generated answers and compares them with the listed features in section 2.4, the Q&A matching features. Based on this comparison it is possible to identify the problem area's and to give a concluding remark about the algorithm.

7.2 Generated answers

As discussed in section 6.3.2, there exist different scenario's within the algorithm to generate answers for all question & answer pairs in the build corpus. All these generated answers should be based on the "topic-comment" pattern, because this pattern has been identified as preferred for the presentation of an answer. To evaluate the generated answers, first needs to be defined what can be seen as a "correct" answer at this stage. A correct answer is at this stage: *a generated answer that is based on the topic-comment order*. This means that regarding grammatical correctness, the explicit use of 'yes/no' and the used formulation nothing is being evaluated. This because the algorithm in only focused on realizing the topic-comment pattern at this stage. Of course the grammatical correctness and the other issues are important, but they should be covered in a future assessment, when the algorithm also tries to cover these problems. This all means that this assessment is oriented at evaluating the topic-comment pattern. To clarify the correctness begin evaluated, some examples of correct generated answers per scenario within the algorithm. These example show that is it possible to generate the topic-comment order, but also illustrate where is goes wrong.

- *** Match found and one subject very likely to be topic ***
  Question: Helpt massage bij RSI?
  Fragment: Betere doorbloeding van je spieren door massage kan (kortdurend) gunstige effecten op je klachten hebben.
  Generated: **massage bij RSI** kan kortdurend gunstige effecten op je klachten hebben.
- *** Match found and one subject very likely to be comment ***
  Question: Wat zijn symptomen van RSI?
  Fragment: Koude , kramp , verkleuring , stijfheid , tinteling , krachtsverlies , en pijn of juist gevoelloosheid zijn allemaal symptomen van RSI.
  Generated: **symptomen van RSI** koude, kramp, verkleuring, stijfheid, tinteling,

krachtsverlies, en pijn, of juist gevoelloosheid.
Wrong: Incorrect use of question topic, missing verb.

- *** Match found but not inside an identified subject ***
  Question: Hoe voorkom je RSI?
  Fragment: We kunnen RSI voorkomen door de spieren en geest te ontspannen , en de bloedsdoorstroming te stimuleren .
  Generated: **voorkom RSI** voorkomen door de spieren en geest te ontspannen en de bloedsdoorstroming te stimuleren.
  Wrong: Incorrect use of question topic, double use of verb.
- *** No match is found, 1 short subject found − > topic ***
  Question: Kunnen oefeningen RSI voorkomen?
  Fragment: Deze stimuleren de bloedsomloop , hebben een ontspannend effect op lichaam en geest , en kunnen klachten en pijn voorkomen.
  Generated: **oefeningen RSI voorkomen** stimuleren de bloedsomloop hebben een ontspannend effect op lichaam, en geest, en kunnen klachten, en pijn, voorkomen.
  Wrong: Incorrect use of question topic, double verb.
- *** No match is found, 1 long subject found − > comment ***
  Question: Waar wordt RSI door veroorzaakt?
  Fragment: Factoren zoals werken in dezelfde houding en het maken van bewegingen zijn de oorzaak van het fysieke syndroom.
  Generated: **RSI door veroorzaakt** factoren zoals werken in dezelfde houding en het maken van bewegingen.
  Wrong: Incorrect use of question topic, missing verb.
- *** No match is found, subjects were found − > shortest is topic ***
  Question: Wat is RSI?
  Fragment: Onderzoek heeft uitgewezen dat repeterende bewegingen , een langdurige statische houding of een combinatie van beiden klachten kunnen veroorzaken.
  Generated: **RSI** heeft uitgewezen dat repeterende bewegingen een langdurige statische houding of een combinatie van beiden klachten kunnen veroorzaken.
  Wrong: Incorrect use of question topic.
- *** No subject and no match is found, total fragment is comment ***
  Question: Wat is de definitie van RSI?
  Fragment: Repetitive Strain Injuries.
  Generated: **de definitie van RSI** Repetitive Strain Injuries.
  Wrong: Incorrect use of question topic, missing verb.

Within these examples, the "answer topic" is indicated by *boldface* to indicate where a repetition of the question topic occurs and the other part of the answer sentence functions as "answer comment". As these examples illustrate, the realization of a "topic-comment" pattern is possible with the implemented algorithm. However, these examples also illustrate that it goes wrong for issues related to grammatical correctness. Although this issue is not covered by the algorithm at this stage, lets take a look at all the generated answers within the corpus in the next section, to provide a good final judgement. Not only regarding the generation of the topic-comment order, but also regarding the grammatical issues as indicated in the examples above.

7.3 Generated answer evaluation

In the corpus there are 31 questions available each accompanied by at least one retrieved fragment. The fragment structure has been altered in different forms to be able to test the robustness of the

algorithm. The following example illustrates this altering of a Dutch text fragment structure in which the sentence part that is indicated in **boldface** should function as the "answer comment" in the end.

- **Koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies, en pijn of juist gevoelloosheid** zijn allemaal symptomen van RSI.
- Bij RSI treden diverse klachten op **zoals koude, kramp, verkleuring, stijfheid, tinteling, en krachtsverlies in handen, schouders, etc.**
- Ze zeggen dat RSI-verschijnselen kunnen **variëren van, koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid.**

By altering the fragments in different forms, it is possible to test that the algorithm is able to identify the 'answer comment' of a text fragment at different places within the fragment. Because, as illustrated in the example above, an 'answer comment' can occur at different places within a retrieved fragment.

Some numbers

Accompanied with the 31 questions are 86 text fragments and using the demonstrator it is possible to generate 86 answers. Most of the answers are not ready yet for an actual presentation of answer text since these are not grammatical correct, but 65% of the generated answers fulfill the topic-comment pattern. These 65% can be seen as nearly ready for an actual presentation when a solution can be found for answer topic revision.

An example of an answer that is almost suitable for presentation, but first needs revision of the answer topic, is the example below. In this example, the answer topic *RSI door veroorzaakt* should be revised into *RSI wordt veroorzaakt door*.

Before: "*RSI door veroorzaakt* factoren zoals ... maken van bewegingen."
After: "*RSI wordt veroorzaakt door* factoren zoals ... maken van bewegingen.

Out of these 65% or 56 correct answers, that comply with topic-comment order, 41 generated answers can be compared to this example and the other 15 can be seen as already 'fluent' (ready for presentation: grammatical correct and based on the topic-comment order) answers. This means that for an appropriate presentation of answer texts, it is definitely necessary to find an algorithm for *answer topic revision*. However, how to realize a good algorithm for answer topic revision is a difficult issue that needs to be covered in future work.

The other 35% cover in total 30 questions, out of which 12 generated answers are based on a wrong identification of answer topic and answer comment and 18 generated answers are totally wrong and hard to identify where it goes wrong within the algorithm. To find a solution for the group of 12 generated answers is related to the correct identification of an 'answer topic' and 'answer comment' within the algorithm. As mentioned in chapter 6 on page 53, the addition of semantic tags could contribute for the three scenario's "* No match is found, 1 short subject found $->$ topic *", "* No match is found, 1 short subject found $->$ topic *" and "* No match is found, subjects were found $->$ shortest is topic *". In these scenario's, the identification process goes wrong too often, because the choices made in these scenario's are based on the word length of subjects and that is not the perfect way to do identification. To improve this process it would be an idea to investigate the use of these semantic tags (*new 'topic/comment' identification*), because it could increase the 65% score into a 79% score of answer that comply with a correct topic-comment order. The 79% is probably a bit optimistic, but an increase of answers that comply with the topic-comment order can be certainly be achieved.

At this point 21 % of the generated answers, would still not comply with a correct topic-comment order after the possible solutions, as described above, would have been implemented. Exactly where it goes wrong and what solutions could contribute to better performance of the algorithm is hard to identify, but is related to issues like: 'topic-fronting' (6), 'incompatible question&answer pair' (8) and 'incorrect algorithm scenario' (4).

When we look at the scenario's responsible for the generated answers, we can present the following overview in table 7.1. The most likely scenarios are activated the most and the three scenarios * No match is found, 1 short subject found $->$ topic *, * No match is found, 1 long subject found $->$ comment * and * No match is found, subjects were found $->$ shortest is topic * are indeed responsible for incorrect answers based on the topic-comment pattern (12, see above, of the 16 answers in total were identified as incorrect).

Scenario	Number	% Total
* Match found and one subject very likely to be topic *	30	35%
* Match found and one subject very likely to be comment *	28	32%
* Match found but not inside an identified subject *	11	13%
* No match is found, 1 short subject found $->$ topic *	8	9%
* No match is found, 1 long subject found $->$ comment *	4	5%
* No match is found, subjects were found $->$ shortest is topic *	4	5%
* No subject and no match is found, total fragment is comment *	1	1%
* Match found but no subject identified *	0	0%

Table 7.1: Number of times a scenario is activated

An overview

Table 7.2 provides an overview of the numbers above and table 7.3 provides the numbers of generated answers that could be realized when the solutions would have been implemented. It is the most optimistic prediction, based on the issues as discussed before. Based on these two tables it is possible to address the problem areas that need attention in future work.

- **Fluent**: These answers are ready for presentation: grammatical correct and based on the topic-comment order.
- **Correct** These answers comply with the topic-comment order.

Description	Number	% Total
Total number answers	86	–
Number of fluent answers	15	17%
Number of not fluent answers	71	83%
Number of correct answers	56	65%
Number of incorrect answers	30	35%

Table 7.2: Assessment outcome

Description with prediction	Number	% Total
Number of correct answers after new 'topic/comment' identification	68	79%
Number of incorrect answers after new 'topic/comment' identification	18	21%
Number of fluent answers after answer topic revision	56	65%
Number of fluent answers after answer topic revision and new identification	68	79%

Table 7.3: Predictions

- The algorithm generates answer that are fluent and ready for the presentation of an answer. However, this number of 17% is to low. Through *answer topic revision* this number of fluent answers can, with an optimistic view, increase to 65%, the total number of answers that are correct answers.
- The number of correct answers is 65% and can, with an optimistic view, be increased to 79% when *new topic/comment identification* is being implemented. In other words, by adding semantical information, it will be possible for the algorithm to do better identification of the answer topic and answer comment. Through better identification, 79% of the generated answers would be in a correct topic-comment order.
- The issues related to 'topic-fronting', 'incompatible question&answer pair' and 'incorrect algorithm scenario' needs thorough investigation of the algorithm to provide suitable solutions. At this stage these issues have not been investigated, through the pressure of time.

7.4 Conclusion

Based on this assessment it becomes clear that the algorithm is able to produce 65% of 'correct' generated answer based on the topic-comment pattern. When looking at fluent answers, only 17% is already suitable for the presentation of answer text. However, to create a better algorithm, with more fluent generated answers, it will be necessary to perform *answer topic revision*. By applying *new 'topic/comment' identification*, the algorithm scenarios , ** No match is found, 1 short subject found − > topic **, ** No match is found, 1 long subject found − > comment ** and ** No match is found, subjects were found − > shortest is topic ** are also likely to produce correct answers based on the topic-comment pattern. So it will be useful to investigate the addition of semantical information to realize a better revision algorithm.

Regarding grammatical correctness, the explicit use of 'yes/no' and the used formulation nothing has been evaluated, since these have not been implemented in the algorithm. They are however important for future development and, as this assessment indicates, important for realizing fluent generated answers. All in all, realizing the topic-comment pattern is possible with the algorithm, but there is still a lot of work to be done for the generation of fluent answer.

Chapter 8

Conclusions & Future work

8.1 Main contributions

For natural presentation of answer text, a foundation has been presented in the form of the topic-comment pattern. This pattern was encountered in literature about information structure theory and confirmed within the performed experiment. All this to realize a congruent answer in which it is clear for a user, what the new contributing information is in relation to the question. By realizing answers based on the topic-comment pattern, the generated answers will be congruent with the question and the new contributing information, is recognizable as the 'answer comment'.

Related to this pattern are the properties of questions & answers, which have been identified as *question type*, *question pattern*, *answer type*, *answer pattern* and *question topic*. Also the main features of congruent question & answer pairs have been tested n the experiment by confirming / disconfirming the hypotheses: *topic-comment linking*, *topic-comment order*, *explicit use of yes/no* and *used formulation*.

The exploration of the aggregation field resulted in the identification different types of aggregation and syntactical aggregation was identified as a candidate for implementation. However, syntactical aggregation has not been implemented in the demonstrator, since the time invested in the revision algorithm took longer then expected. The information presented in this research shows where and how aggregation can be applied using syntactic information and forms a good starting point for future work.

The algorithm that is responsible for realizing the topic-comment pattern, has been evaluated and it showed that the algorithm is capable of realizing answers based on the topic-comment pattern. However, the assessment showed the algorithm works by no means without errors. Two issues related to question topic revision and better identification with the algorithm have been indicated as interesting point of departure for future work and are likely to contribute to the overall performance of the algorithm. However, these are pure speculations and need investigation. At this stage the generated answers cannot be defined as natural generated, but 65% can be defined as correct generated answers regarding the topic-comment pattern.

8.2 Future work

As this research was a first attempt within the IMOGEN project to investigate means text revision to generated congruent question & answer pair, a lot of future work has been identified.

Revision of question topic At the moment the question topic is used as identified by the Analyzer. However, most of the time the form of this phrase cannot be used fluently in the generated answer sentence. Further investigation in means of revising the question topic is necessary to create fluent generated answer sentences.

Addition of semantical information As described in the assessment, chapter 7, the addition of extra semantical information in the form of tags could contribute to a better revision algorithm. Especially for three of the possible scenario's within the algorithm where at this stage the 'answer topic' and 'answer comment' are identified by the word length of identified subjects and that is not the perfect way to do identification.

Aggregation The use of syntactical clause aggregation was investigated in this research and seems promising. Especially for text fragments that are not yet in an aggregated form or for text fragments that need to be revised for a Text-To-Speech module it would contribute to a better generated answer. However, in which form of text fragments to apply it, has to be investigated more thoroughly.

Larger retrieved text fragments At this stage only sentence long retrieved material is used and covered inside the algorithm. However, to provide an answer presentation that covers completer information it will be necessary to identify the "topic-comment" pattern through other means that are focused on multiple sentences.

Multi-fragment fusion This is closely related to the above point, in the case that completer information needs to be presented. When fusion occurs a fragment changes in contents and the identification of an 'answer topic' and 'answer comment' becomes probably more difficult. Investigation into this issue and the use of the topic-comment pattern will be necessary.

Multi-modal constraints An thorough investigation into the effects that multi-modal means of presentation have on each other. When using images next to text for example, the use of text may become unnecessary. How these different means of presentation have influence on each other, needs to be investigated.

Postface

I started the phase step of my degree with the intension to work on an issue that got my attention during my internship at British Telecom: the final step within an Question answering system that is focused at presenting the answer. Because I got the feeling that the presentation of an answer is as important for the satisfaction of an user as the actual answer itself. I got in touch with Mariët Theune and Rieks op den Akker and heard about the IMOGEN project that is focused on Interactive Multimodal Output Generation. This project also includes the revision of text and I delimited it especially towards generating an answer that is in agreement with the question.

The first phases of my approach went rather well and finishing my research proposal, building a suitable corpus, realizing the literature survey's about question & answer matching and aggregation, and setting up the experiment to confirm / disconfirm the stated hypotheses, were finished before I knew. During these phases I came across a lot of interesting papers and it sometimes was hard to identify the usable and non-usable papers. What I did discover, was that my original idea of syntactical revision was not broad enough compared to generation systems out of the Natural Language Generation field. As such I had to focus myself on one particular problem: the identification of an underlying pattern for natural presentation of answer texts.

When these hypotheses were confirmed and disconfirmed in the performed experiment, it was time to come up with an algorithm to cover these hypotheses. Looking at the time still left to complete my thesis within a reasonable amount of time, it became clearly that it would be impossible to cover all the hypotheses problems in this research. As such, I choose for the realization of the "topic-comment" pattern as underlying pattern for the presentation of answer texts and left the other issues for future work. These issues, definitely need further investigation in the future, since the problems that arose in the assessment pointed exactly at these issues. To be able to make theses choices is rather difficult, but necessary in order to complete a master thesis within a reasonable amount of time.

As such, I was not able to come up with a module that could generate a natural presentation of answer texts in the end. However, I was able to come up with an algorithm that is capable of realizing the underlying "topic-comment" pattern and that can be used for further research within the TKI-group at the University of Twente.

All in all it was an unique experience and a good way to finish my study at the University of Twente, a period never to forget.

Bibliography

[Anaya and Kosseim, 2003] Anaya, G. and Kosseim, L. (2003). Generation of natural responses through syntactic patterns. In *Traitement Automatique des Langues Naturelles (TALN)*, Batz-sur-Me.

[Bouma and Kloosterman, 2002] Bouma, G. and Kloosterman, G. (2002). Querying dependency treebanks in xml. In *Proceedings of the Third international conference on Language Resources and Evaluation (LREC)*, Gran Canaria.

[Bouma et al., 2001] Bouma, G., van Noord, G., and Malouf, R. (2001). Alpino: Wide-coverage computational analysis of dutch. In *Computational Linguistics in The Netherlands 2000*. Rodopi, Amsterdam/NY.

[Campbell, 2000] Campbell, I. (2000). *The ostensive model of developing information-needs*. PhD thesis, University of Glasgow.

[Carroll et al., 1998] Carroll, J., Briscoe, T., and Sanfilippo, A. (1998). Parser evaluation: A survey and a new proposal. In *Proceedings of the first International Conference on Language Resources and Evaluation (LREC)*, pages 447–454, Granada, Spain.

[Cooper, 1996] Cooper, R. (1996). Head-driven phrase structure grammar. In Brown, K. and Miller, J., editors, *The Concise Encyclopedia of Syntactic Theories*, pages 191–196, Oxford. Pergamon.

[Dikkers, 2003] Dikkers, E. (2003). Question & answering. England, Ipswich. Internship at British Telecom (Language Technology Group).

[Engel, 1996] Engel, U. (1996). Tesnière mißverstanden. In Gréciano, G. and Schumacher, H., editors, *Lucien Tesnière - Syntaxe structurale et opérations mentales*, volume 348 of *Linguistische Arbeiten*, pages 53–61, Tubingen. Max Niemeyer Verlag.

[Grosz et al., 1995] Grosz, B., Joshi, A. K., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. In *Computational Linguistics*, volume 21, pages 203–226.

[Haeseryn et al., 1997] Haeseryn, W., Romijn, K., Geerts, G., de Rooij, J., and v/d Toorn, M. (1997). *Algemene Nederlandse Spraakkunst*. Martinus Nijhof (Groningen) and Wolters Plantyn (Deurne), tweede, geheel herziene druk edition.

[Hajicova et al., 1998] Hajicova, E., Panevova, J., and Sgall, P. (1998). Language re-sources need annotations to make them really reusable:the prague dependency treebank. In *proceedings of LREC 1998*, pages 713–718, Granada, Spain.

[Halliday, 1985] Halliday, M. (1985). *An introduction to functional grammar*. Arnold, London.

[Harold, 2001] Harold, E. R. (2001). *Processing XML with Java: A guide to SAX, DOM, JDOM, JAXP and TrAX*. http://www.cafeconleche.org/books/xmljava/. Addison-Wesley.

[Hays, 1964] Hays, D. (1964). Dependency theory: A formalism and some observations. *Language*, (40):511–525.

[Hiz, 1978] Hiz, H. (1978). Introduction. In Hiz, H., editor, *Questions*, pages IX–XVII, Dordrecht. Reidel.

[Hovy et al., 2002] Hovy, E., Hermjakob, U., and Ravichandran, D. (2002). Learning surface text patterns for a question answering system. In *Proceedings of ACL*.

[Hudson, 1991] Hudson, R. (1991). *English Word Grammar*. Basil Blackwell, Cambridge, MA.

[Hudson, 1996] Hudson, R. (1996). Word grammar. In Brown, K. and Miller, J., editors, *Concise Encyclopedia of Syntactic Theories*, pages 368–372. Oxford: Elsevier.

[Kosseim et al., 2003] Kosseim, L., Plamondon, L., and Guillemette, L. (2003). Answer formulation for question-answering. In Halifax, Y. X. and Chaib-draa, B., editors, *Proceedings of The Sixteenth Conference of the Canadian Society for Computational Studies of Intelligence*, pages 24–34, Canada. Springer-Verlag.

[Krifka, 2001] Krifka, M. (2001). For a structured meaning account of questions and answers. In Fery, C. and Sternefeld, W., editors, *Audiatur Vox Sapientia. A Festschrift for Arnim von Stechow, Akademie Verlag (= studia grammatica 52)*, pages 287–319, Berlin.

[Kuboň, 1999] Kuboň, P. (1999). *Information Packaging Revisited*. PhD thesis, Simon Fraser University, Canada.

[Lin, 1997] Lin, C. (1997). *Robust automated topic identification*. PhD thesis, University of Southern California.

[Lin et al., 2003] Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., and Karger, D. (2003). What makes a good answer? the role of context in question answering. In *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction*, Zurich, Switzerland.

[Mann and Thompson, 1987] Mann, W. and Thompson, S. (1987). Rhetorical structure theory: A theory of text organization. Technical report isi/rs-87-190, ntis identifying number ada 183038, University of Southern California, Information Sciences Institute, Marina del Rey, CA, USA.

[Melčuk, 1988] Melčuk, I. (1988). *Dependency Syntax: Theory and Practice*. State University of New York Press, New York.

[Moortgat et al., 2002] Moortgat, M., Schuurman, I., and v/d Wouden, T. (2002). Cgn syntactische annotatie.

[Negri et al., 2003] Negri, M., Magnini, B., and Tanev, H. (2003). Answer-driven reasoning as a framework for intelligent information access. In *Proceedings of AI*IA - Eight National Congress of Associazione Italiana per l'Intelligenza Artificiale*, Pisa, Italy.

[on Language Engineering Standards, 1996] on Language Engineering Standards, E. A. G. (1996). Recommendations for the morphosyntactic annotation of corpora. *EAGLES Document EAG - TCWG - MAC/R*.

[Rambow and Joshi, 1994] Rambow, O. and Joshi, A. (1994). A formal look at dependency grammars and phrase-structure grammars, with special consideration of word order phenomena. In Wanner, L., editor, *Current Issues in Meaning-Text Theory*, London. Pinter.

[Reape and Mellish, 1999] Reape, M. and Mellish, C. (1999). Just what is aggregation anyway? In *the 7th European Workshop on Natural Language Generation*, Toulouse, France.

[Reiter and Dale, 1997] Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Journal of Natural Language Engineering*, 3(1):57–87.

[Schneider, 1998] Schneider, G. (1998). A linguistic comparison constituency, dependency, and link grammar. Master's thesis, University of Zurich.

[Shaw, 2002] Shaw, J. (2002). *Clause Aggregation: An approach to generating concise text*. PhD thesis, Columbia University.

[Stokhof and Groenendijk, 1997] Stokhof, M. and Groenendijk, J. (1997). Questions. In van Benthem, J. and ter Meulen, A., editors, *Handbook of Logic and Language*, pages 1055–1124. Amsterdam/Cambridge, Mass., Elsevier/MIT Pess, Amsterdam/Cambridge, Mass.

[Tesnière, 1959] Tesnière, L. (1959). *Eléments de syntaxe structurale*. Librairie Klincksieck, Paris.

[Vallduvi, 1993] Vallduvi, E. (1993). Information packaging: A survey. Technical report, Centre for Cognitive Science and Human Communication Research Centre, University of Edinburgh. Report prepared for Word Order, Prosody, and Information Structure.

[v/d Beek et al., 2002a] v/d Beek, L., Bouma, G., and v Noord., G. (2002a). Annotation differences between cgn and the alpino treebank. Internal rapport.

[v/d Beek et al., 2002b] v/d Beek, L., Bouma, G., and van Noord, G. (2002b). Een brede computationele grammatica voor het Nederlands. *Nederlandse Taalkunde*.

[v/d Wouden et al., 2002] v/d Wouden, T., Hoekstra, H., Moortgat, M., Renmans, B., and Schuurman, I. (2002). Syntactische annotatie voor het corpus gesproken nederlands (cgn). In *Nederlandse Taalkunde*, volume 4, pages 335–352.

[Verschuren and Doorewaard, 2000] Verschuren, P. and Doorewaard, H. (2000). *Het ontwerpen van een onderzoek*. Uitgeverij Lemma BV, Utrecht.

[von Stechow and Zimmerman, 1984] von Stechow, A. and Zimmerman, E. (1984). Term answers and contextual change. In *Linguistics*, volume 22, pages 3–40.

[Wilcock, 2001] Wilcock, G. (2001). Pipelines, templates and transformations: Xml for natural language generation. In *Proceedings of the First NLP and XML Workshop*, NLPRS-2001, pages 1–8, Tokyo.

[Wilkinson, 1995] Wilkinson, J. (1995). Aggregation in natural language generation: Another look. Technical report, Department of Computer Science, University of Waterloo. Co-op work term report.

Appendix A

Question-Answer experiment

Voor mijn afstuderen, doe ik onderzoek naar de aspecten die een rol spelen bij het zo natuurlijk mogelijk aanbieden van een antwoord. Een vraag & antwoord, vormen een "natuurlijk" paar, maar wat is "natuurlijk"? Aan de hand van deze evaluatie, verwacht ik inzicht te krijgen in wat een grote groep mensen (waaronder jij dus) als een natuurlijk antwoord op een vraag ervaart. Deze evaluatie bestaat uit 22 vragen en per vraag kan worden aangegeven welk vorm van het antwoord, het meest "natuurlijk" (in jouw ogen) op de vraag aansluit. De antwoorden zijn gebaseerd op teksten van Internet die gaan over RSI en de inhoud mag als correct worden aangenomen. De mogelijke antwoorden zullen veel op elkaar lijken, maar toch onderling subtiel te onderscheiden zijn. Als je de meest "natuurlijke" keuze wilt aangeven in het vakje onder de vragen en vervolgens dit document voor 21 juni wilt mailen naar: e.j.w.dikkers@student.utwente.nl, dan zou je mij een groot plezier doen.

Alvast bedankt,

Erik Dikkers

Naam:	
Leeftijd:	
Opleiding:	
Plaats:	

* Deze gegevens kunnen een bepaalde trend aangegeven. Bijvoorbeeld dat personen die een bepaalde opleiding als achtergrond hebben, een bepaalde antwoordstijl verwachten of dat een antwoordstijl leeftijdsgebonden is. De evaluatie kan uiteraard ook anoniem worden ingevuld.

Vraag 1: Wat is RSI?
Mogelijke antwoorden:

A) Een verzamelnaam voor klachten, symptomen en syndromen die voorkomen in bovenrug, nek- en schoudergebied, armen, ellebogen, polsen, handen en vingers, dat is RSI.

B) RSI is een verzamelnaam voor klachten, symptomen en syndromen die voorkomen in bovenrug, nek- en schoudergebied, armen, ellebogen, polsen, handen en vingers.

C) Een verzamelnaam voor klachten, symptomen en syndromen die voorkomen in bovenrug, nek- en schoudergebied, armen, ellebogen, polsen, handen en vingers.

Vraag 2: Is RSI te genezen?
Mogelijke antwoorden:

A) Ja, RSI is te genezen, mits je er maar vroeg genoeg bij bent.

B) RSI is te genezen, mits je er maar vroeg genoeg bij bent.

C) Ja, mits je er maar vroeg genoeg bij bent, is RSI te genezen.

D) Mits je er maar vroeg genoeg bij bent, is RSI te genezen.

Vraag 3: Wat is de definitie van RSI?
Mogelijke antwoorden:

A) De definitie van RSI is Repetitive Strain Injuries. Het is een verzamelnaam voor klachten aan nek, schouder, bovenrug, arm, elleboog en pols of hand, die tot beperkingen of uitval leiden.

B) Repetitive Strain Injuries. Het is een verzamelnaam voor klachten aan nek, schouder, bovenrug, arm, elleboog en pols of hand, die tot beperkingen of uitval leiden.

C) Repetitive Strain Injuries is de definitie van RSI. Het is een verzamelnaam voor klachten aan nek, schouder, bovenrug, arm, elleboog en pols of hand, die tot beperkingen of uitval leiden.

Vraag 4: Wat is een goede behandeling bij RSI?
Mogelijke antwoorden:

A) Uit ervaring blijkt dat de beste behandeling een brede aanpak is. Om die reden adviseert RSI-patiëntenvereniging een integrale benadering.

B) Een brede aanpak, dat blijkt uit ervaring, is de beste behandeling. Om die reden adviseert de RSI-patiëntenvereniging een integrale benadering.

C) Uit ervaring blijkt, een brede aanpak. Om die reden adviseert de RSI-patiëntenvereniging een integrale benadering.

Vraag 5: Kan RSI verdwijnen?
Mogelijke antwoorden:

A) Met adequate maatregelen is het mogelijk om RSI-klachten binnen enkele weken te laten verdwijnen en herhaling te voorkomen. Zonder maatregelen, kunnen de klachten snel in ernst toenemen en zich uitbreiden.

B) Binnen enkele weken RSI-klachten te laten verdwijnen en herhaling te voorkomen, is mogelijk met adequate maatregelen. Zonder maatregelen, kunnen de klachten snel in ernst toenemen en zich uitbreiden.

C) Ja, met adequate maatregelen is het mogelijk om RSI-klachten binnen enkele weken te laten verdwijnen en herhaling te voorkomen. Zonder maatregelen, kunnen de klachten snel in ernst toenemen en zich uitbreiden.

D) Ja, binnen enkele weken RSI-klachten te laten verdwijnen en herhaling te voorkomen, is mogelijk met adequate maatregelen. Zonder maatregelen, kunnen de klachten snel in ernst toenemen en zich uitbreiden.

Vraag 6: Wat zijn risicofactoren voor RSI?
Mogelijke antwoorden:

A) Zware lichamelijke belasting (repeterende bewegingen, kracht, statische belasting, extreme houdingen, trillingen), maar liggen ook op het psychosociale vlak (zoals werkdruk, spanningen en slechte sfeer op het werk (NRC, 2001).

B) Risicofactoren voor RSI kunnen zowel op werk- als privé-gebied liggen. Risicos komen voort uit zware lichamelijke belasting (repeterende bewegingen, kracht, statische belasting, extreme houdingen, trillingen), maar liggen ook op het psychosociale vlak (zoals werkdruk, spanningen en slechte sfeer op het werk (NRC, 2001).

C) Zowel op werk- als privé-gebied kunnen risicofactoren voor RSI liggen. Risicos komen voort uit zware lichamelijke belasting (repeterende bewegingen, kracht, statische belasting, extreme houdingen, trillingen), maar liggen ook op het psychosociale vlak (zoals werkdruk, spanningen en slechte sfeer op het werk (NRC, 2001).

Vraag 7: Waardoor ontstaat rsi?
Mogelijke antwoorden:

A) Door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI), ontstaat RSI grofweg. Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

B) RSI ontstaat grofweg door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

C) Door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI).

Vraag 8: Wat zijn oorzaken van rsi?
Mogelijke antwoorden:

A) Oorzaken van RSI zijn: eenzijdigheid van werkzaamheden, lange duur van bepaalde werkzaamheden, statische houdingen, onvoldoende pauzes en korte onderbrekingen (micropauzes), overbodige spierspanningen door onjuiste werksituaties.

B) Eenzijdigheid van werkzaamheden, lange duur van bepaalde werkzaamheden, statische houdingen, onvoldoende pauzes en korte onderbrekingen (micropauzes), overbodige spierspanningen door onjuiste werksituaties, zijn oorzaken van RSI.

C) Eenzijdigheid van werkzaamheden, lange duur van bepaalde werkzaamheden, statische houdingen, onvoldoende pauzes en korte onderbrekingen (micropauzes) en overbodige spierspanningen door onjuiste werksituaties.

Vraag 9: Welke factoren leiden tot RSI?
Mogelijke antwoorden:

A) Overmatig uitoefenen van kracht, werken in ongemakkelijke houdingen, voortdurend werken in dezelfde houding (statische belasting) en repeterende bewegingen zijn onder meer risicofactoren voor RSI.

B) Factoren die leiden tot RSI zijn onder meer overmatig uitoefenen van kracht, werken in ongemakkelijke houdingen, voortdurend werken in dezelfde houding (statische belasting) en repeterende bewegingen.

C) Overmatig uitoefenen van kracht, werken in ongemakkelijke houdingen, voortdurend werken in dezelfde houding (statische belasting) en repeterende bewegingen.

Vraag 10: Helpt massage bij RSI?
Mogelijke antwoorden:

A) Ja, massage verbetert de doorbloeding van je spieren, wat (kortdurend) gunstige effecten op je klachten kan hebben.

B) Massage verbetert de doorbloeding van je spieren, wat (kortdurend) gunstige effecten op je klachten kan hebben.

C) Massage helpt de doorbloeding van je spieren, wat (kortdurend) gunstige effecten op je klachten kan hebben.

D) Ja, massage helpt de doorbloeding van je spieren, wat (kortdurend) gunstige effecten op je klachten kan hebben.

Vraag 11: Hoe kan RSI ontstaan?
Mogelijke antwoorden:

A) RSI kan grofweg ontstaan door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

B) Door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI), kan RSI grofweg ontstaan. Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

C) Door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI).

Vraag 12: Hoe voorkom je RSI?
Mogelijke antwoorden:

A) Wat we kunnen doen heeft vrijwel allemaal te maken met het ontspannen van de spieren en geest, stimuleren van de bloedsdoorstroming.

B) Voor het voorkomen van RSI is geen standaardoplossing aan te dragen. Wat we kunnen doen heeft vrijwel allemaal te maken methet ontspannen van de spieren en geest, stimuleren van de bloedsdoorstroming.

C) Er is geen standaardoplossing aan te dragen voor het voorkomen van RSI. Wat we kunnen doen heeft vrijwel allemaal te maken met het ontspannen van de spieren en geest, stimuleren van de bloedsdoorstroming.

Vraag 13: Kunnen oefeningen RSI voorkomen?
Mogelijke antwoorden:

A) Korte oefeningen tijdens het werk (maar ook daarbuiten) kunnen klachten en pijn voorkomen. Ze stimuleren de bloedsomloop en hebben een ontspannend effect op lichaam en geest.

B) Ja, korte oefeningen tijdens het werk (maar ook daarbuiten) kunnen klachten en pijn voorkomen. Ze stimuleren de bloedsomloopen hebben een ontspannend effect op lichaam en geest.

C) Ja, tijdens het werk (maar ook daarbuiten) kunnen korte oefeningen klachten en pijn voorkomen. Ze stimuleren de bloedsomloop en hebben een ontspannend effect op lichaam en geest.

D) Tijdens het werk (maar ook daarbuiten) kunnen korte oefeningen klachten en pijn voorkomen. Ze stimuleren debloedsomloop en hebben een ontspannend effect op lichaam en geest.

Vraag 14: Welke factoren zijn verantwoordelijk voor RSI?
Mogelijke antwoorden:

A) Een hoge werkdruk, stress en het langdurig dezelfde handelingen uit (moeten) voeren, zijn de belangrijkste factoren die het ontstaan van RSI bevorderen.

B) Een hoge werkdruk, stress en het langdurig dezelfde handelingen uit (moeten) voeren.

C) De belangrijkste factoren die het ontstaan van RSI bevorderen zijn een hoge werkdruk, stress en het langdurig dezelfde handelingen uit (moeten) voeren.

Vraag 15: Waar wordt RSI door veroorzaakt?
Mogelijke antwoorden:

A) Een combinatie van fysieke factoren, waaronder werken in dezelfde houding en het maken van bewegingen, daar wordt RSI door veroorzaakt. Niet fysieke factoren blijken de kans op RSI te benvloeden.

B) RSI wordt veroorzaakt door een combinatie van fysieke factoren, waaronder werken in dezelfde houding en het maken van bewegingen. Niet fysieke factoren blijken de kans op RSI te benvloeden.

C) Door een combinatie van fysieke factoren, waaronder werken in dezelfde houding en het maken van bewegingen. Niet fysieke factoren blijken de kans op RSI te benvloeden.

Vraag 16: Wat is een goede behandeling voor rsi?
Mogelijke antwoorden:

A) Gedragsverandering en een multidisciplinaire aanpak van de klachten en oorzaken.

B) Gedragsverandering en een multidisciplinaire aanpak van de klachten en oorzaken zijn een behandeling van RSI.

C) Een behandeling van RSI bestaat vooral uit gedragsverandering en een multidisciplinaire aanpak van de klachten en oorzaken.

Vraag 17: Wat zijn symptomen van RSI?
Mogelijke antwoorden:

A) Koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid zijn symptomen van RSI. Ze kunnen zich voordoen in nek, schouders, armen, polsen, handen en in heel uitzonderlijke gevallen de benen.

B) De symptomen van RSI zijn uiteenlopend van aard en kunnen variëren van, koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid. Ze kunnen zich voordoen in nek, schouders, armen, polsen, handen en in heel uitzonderlijke gevallen de benen.

C) Koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid. Ze kunnen zich voordoen in nek, schouders, armen, polsen, handen en in heel uitzonderlijke gevallen de benen.

Vraag 18: Zijn er medicijnen voor RSI?
Mogelijke antwoorden:

A) Nee, er is helaas geen medicijn. Er zijn echter wel medicijnen die de symptomen kunnen bestrijden.

B) Nee, voor RSI is er helaas geen medicijn. Er zijn echter wel medicijnen die de symptomen kunnen bestrijden.

C) Er is helaas geen medicijn. Er zijn echter wel medicijnen die de symptomen kunnen bestrijden.

D) Voor RSI is er helaas geen medicijn. Er zijn echter wel medicijnen die de symptomen kunnen bestrijden.

Vraag 19: Waardoor ontstaat rsi?
Mogelijke antwoorden:

A) RSI begint grofweg door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

B) RSI kan grofweg ontstaan door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling(statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

C) RSI ontstaat grofweg door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

Vraag 20: Waar wordt RSI door veroorzaakt?
Mogelijke antwoorden:

A) RSI wordt veroorzaakt door een combinatie van fysieke factoren, waaronder werken in dezelfde houding en het maken van bewegingen. Niet fysieke factoren blijken de kans op RSI te benvloeden.

B) RSI komt tot uiting door een combinatie van fysieke factoren, waaronder werken in dezelfde houding en het maken van bewegingen. Niet fysieke factoren blijken de kans op RSI te benvloeden.

C) RSI ontstaat door een combinatie van fysieke factoren, waaronder werken in dezelfde houding en het maken van bewegingen. Niet fysieke factoren blijken de kans op RSI te benvloeden.

Vraag 21: Welke factoren zijn verantwoordelijk voor RSI?
Mogelijke antwoorden:

A) De belangrijkste factoren die het ontstaan van RSI bevorderen zijn een hoge werkdruk, stress en het langdurig dezelfde handelingen uit (moeten) voeren.

B) Verantwoordelijk voor RSI zijn een hoge werkdruk, stress en het langdurig dezelfde handelingen uit (moeten) voeren.

C) De belangrijkste factoren voor RSI zijn een hoge werkdruk, stress en het langdurig dezelfde handelingen uit (moeten) voeren.

Vraag 22: Hoe kan RSI ontstaan?
Mogelijke antwoorden:

A) RSI ontstaat grofweg door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

B) RSI kan grofweg ontstaan door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

C) RSI begint grofweg door het maken van steeds dezelfde bewegingen (dynamische RSI) en door gebrek aan afwisseling (statische RSI). Door de sterke toename van het gebruik van computers zien we tegenwoordig steeds vaker de statische vorm van RSI.

Deze vraag 22 vormt het einde van de evaluatie. Mocht je nog opmerkingen hebben over een bepaalde vraag of algemener over de evaluatie, dan kan dat hier:

Nogmaals bedankt voor het invullen van deze evaluatie en het meewerken aan mijn afstudeer onderzoek. Het liefst voor 21 juni per mail terug sturen aan: e.j.w.dikkers@student.utwente.nl

Appendix B

Alpino identified relations

Node tags

-1	UNKNOWN	<automatisch>
0	--	<unbound>
1	SMAIN	declaratieve zin (V2)
2	SSUB	bijzin (V-finaal)
3	SV1	zin met V op de eerste plaats
4	INF	kale-infinitiefgroep
5	PPART	voltooid/passief-deelwoordgroep
6	PPRES	tegenwoordig-deelwoordgroep
7	CP	zinsdeel ingeleid door onderschikkend vw. of vw./vz. v. verg.
8	--	<dummy>
9	MWU	merged-word-unit ("drie en twintig", "Jan van den Berg")
10	TI	te-infinitiefgroep
11	OTI	om-te-infinitiefgroep
12	AHI	aan-het-infinitiefgroep
13	ADVP	bijwoordgroep (alleen voor echte bijwoorden)
14	DETP	determinatorgroep ("bijna alle" in "bijna alle boeken")
15	AP	adjectiefgroep (ook voor adverbiaal gebruikte adjectieven)
16	PP	prepositiegroep
17	NP	nominale groep
18	SVAN	van-zin (complement in directe rede)
19	REL	relatiefzin
20	WHREL	hoofdloze relatiefzin
21	WHQ	constituentvraag: hoofdzin
22	WHSUB	constituentvraag: bijzin
23	CONJ	conjunctie
24	DU	discourse-unit (asyndetische constructie)
25	LIST	asyndetische conjunctie
26	COMPP	zinsdeel met 'meer' of 'even' als hoofd en CP als complement

Edge tags

-1	UNKNOWN	<automatisch>
0	--	<unbound>
1	HD	hoofd
2	HDF	staart (scheidbaar deel) van circumpositie
3	DET	determinator
4	PART	partitief
5	SU	subject, onderwerp
6	SUP	voorlopig subject
7	OBJ1	direct object van V, (eerste) complement van P, A, N
8	POBJ1	voorlopig OBJ1
9	OBJ2	secundair object (IO, EO, BO)
10	--	<dummy>
11	SE	verplicht reflexief object
12	SVP	scheidbaar deel van werkwoord
13	PREDC	predicatief complement
14	PC	voorzetselvoorwerp
15	VC	verbaal complement, beknopte bijzin
16	LD	locatief of directioneel complement
17	ME	maat(/duur/gewicht)-complement
18	CMP	complementeerder/hoofd van CP, SVAN, TI, OTI of AHI
19	RHD	complementeerder/hoofd van (hoofdloze) relatiefzin
20	WHD	complementeerder/hoofd van WHQ of WHSUB
21	BODY	romp van CP, SVAN, TI, OTI, AHI, REL, WHQ of WHSUB
22	PREDM	bepaling v. gesteldheid 'tijdens de handeling'
23	MOD	algemeen label voor bepaling/modificeerder
24	CRD	nevenschikker
25	CNJ	lid van nevenschikking
26	NUCL	kernzin (in DU)
27	SAT	satelliet: aan- of uitloop (in DU) met binding in NUCL
28	TAG	aanhangsel, voor- of tussenvoegsel
29	DP	elk der delen van een DU
30	PRT	elk der delen van een partikelgroep
31	OBCOMP	vergelijkingscomplement (compl. van 'meer'/'even')
32	APPOS	bijstelling
33	LP	elk der delen van een LIST
34	DLINK	"en", "maar", "want" o.i.d. aan het begin van een uiting
35	MWP	elk der delen van een MWU

Appendix C

Algorithm scenarios

Question 2: "Wat zijn symptomen van RSI ?"
Fragment 2: "Ze zeggen dat RSI-verschijnselen kunnen variëren van, koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid en kunnen zich voordoen in nek schouders, armen, polsen, handen en in heel uitzonderlijke gevallen de benen."

Answer 2: "[symptomen van RSI] kunnen variëren van koude, kramp, verkleuring, stijfheid, tinteling, krachtsverlies tot pijn of juist gevoelloosheid en kunnen zich voordoen in nek, schouders, armen polsen, handen, en in heel uitzonderlijke gevallen de benen."

Identified properties:

Question type: **Wh-word**
Based on *whq*-relation and the word "Wat" in the dependency tree.
Question topic: **'symptomen van RSI'**
Based on *NounPhrase*-relation in the dependency tree.
Answer type: **(Object/Idea/Action)**
Based purely on the identified wh-word 'Wat'.
Fragment subject: **'Ze & RSI-verschijnselen,**
Based on identified *su*-relations in the dependency tree.

Revision algorithm:

```
1.  Match found for ''RSI-verschijnselen''.
2.  Two fragment subjects found and one with highest score.
    Scenario: * Match found and one subject very likely to be topic *.
3.  IdentifiedTopic is known with index from ''RSI-verschijnselen''.
        Insert ''symptomen van RSI'' at identifiedTopic index.
        Start Cleaning of ''Ze zeggen dat'' necessary.
4.  No topic fronting, topic already placed in front segment on index 4.
```

Question 3: "Hoe voorkom je RSI ?"
Fragment 3: "We kunnen RSI voorkomen door de spieren en geest te ontspannen, en de bloedsdoorstroming te stimuleren."

Answer 3: "We kunnen [voorkom RSI] voorkomen door de spieren en geest te ontspannen en de bloedsdoorstroming te stimuleren."

Identified properties:

Question type: **Wh-word**
Based on *whq*-relation and the word "Hoe" in the dependency tree.
Question topic: **'voorkom RSI'**
Based on *verb,name*-relation in the dependency tree.
Answer type: **Manner**
Based purely on the identified wh-word 'Hoe'.
Fragment subject: **'We'**
Based on identified *su*-relation in the dependency tree.

Revision algorithm:

```
1.  Match found for ''RSI''.
2.  One fragment subject found with no increased score.
    Scenario: * Match found but not inside an identified subject *.
3.  IdentifiedTopic is known with index from ''RSI''.
        Insert ''voorkom RSI'' at identifiedTopic index.
        No cleaning necessary.
4.  No topic fronting, topic already placed in front segment on index 3.
```

Question 4: "Kunnen oefeningen RSI voorkomen?"
Fragment 4: "Deze stimuleren de bloedsomloop, hebben een ontspannend effect op lichaam en geest, en kunnen klachten en pijn voorkomen."

Answer 4: "[oefeningen RSI voorkomen] stimuleren de bloedsomloop hebben een ontspannend effect op lichaam, en geest, en kunnen klachten, en pijn, voorkomen."

Identified properties:

Question type: **Yes/No**
Based on *sv1*-relation and the verb "Kunnen" in the dependency tree.
Question topic: **'oefeningen RSI voorkomen'**
Based on *verb,noun,name*-relation in the dependency tree.
Answer type: **Boolean**
Based purely on the identified question type.
Fragment subject: **'Deze'**
Based on identified *su*-relation in the dependency tree.

Revision algorithm:

```
1.  No match is found based on the constraints.
2.  One fragment subject found.
    Scenario: * No match is found, 1 short subject found -> topic *.
3.  IdentifiedTopic is known with index from ''Deze''.
        Insert ''oefeningen RSI voorkomen'' at identifiedTopic index.
        No cleaning necessary.
4.  No topic fronting, topic already placed in front segment on index 1.
```

Question 5: "Waar wordt RSI door veroorzaakt?"
Fragment 5: "Factoren zoals werken in dezelfde houding en het maken van bewegingen zijn de oorzaak van het fysieke syndroom."

Answer 5: "[RSI door veroorzaakt] factoren zoals werken in dezelfde houding en het maken van bewegingen."

Identified properties:

Question type: **WH-word**
Based on *whq*-relation and the word "Waar" in the dependency tree.
Question topic: **'RSI door veroorzaakt'**
Based on *name,prep,verb*-relation in the dependency tree.
Answer type: **Place**
Based purely on the identified question type.
Fragment subject: **'Factoren zoals werken . . . maken van bewegingen'**
Based on identified *su*-relation in the dependency tree.

Revision algorithm:

```
1.  No match is found based on the constraints.
2.  One fragment subject found > 3.
    Scenario: * No match is found, 1 long subject found -> comment *.
3.  IdentifiedComment is known with index from 'Factoren till bewegingen'.
        Insert ''RSI door veroorzaakt'' on top.
        Start cleaning of ''zijn de oorzaak van het fysieke syndroom''.
4.  No topic fronting, topic already placed in front segment.
```

Question 6: "Wat is RSI?"
Fragment 6: "Onderzoek heeft uitgewezen dat repeterende bewegingen , een langdurige statische houding of een combinatie van beiden klachten kunnen veroorzaken"

Answer 6: "RSI heeft uitgewezen dat repeterende bewegingen een langdurige statische houding of een combinatie van beiden klachten kunnen veroorzaken."

Identified properties:

Question type: **WH-word**
Based on *whq*-relation and the word "Wat" in the dependency tree.
Question topic: **'RSI '**
Based on *name*-relation in the dependency tree.
Answer type: **(Object/Idea/Action)**
Based purely on the identified question type.
Fragment subject: **'Onderzoek & repeterende bewegingen een . . . combinatie van beiden'**
Based on identified *su*-relations in the dependency tree.

Revision algorithm:

```
1.  No match is found based on the constraints.
2.  Two fragment subject found with no increased score.
    Scenario: * No match is found, subjects were found -> shortest is topic *.
3.  IdentifiedTopic is known with index of 'Onderzoek'.
        Insert ''RSI'' at IdentifiedTopic index.
        No cleaning necessary.
4.  No topic fronting, topic already placed in front segment at index 1.
```

Question 7: "Wat is de definitie van RSI?"
Fragment 7: "Repetitive Strain Injuries"

Answer 7: "[de definitie van RSI] Repetitive Strain Injuries."

Identified properties:

Question type: **WH-word**
Based on *whq*-relation and the word "Wat" in the dependency tree.
Question topic: **'de definitie van RSI'**
Based on *det,noun,prep,name*-relation in the dependency tree.
Answer type: **(Object/Idea/Action)**
Based purely on the identified question type.
Fragment subject: **'...'**
No identified *su*-relation in the dependency tree.

Revision algorithm:

```
1.  No match is found based on the constraints.
2.  No subjects found.
    Scenario: * No subject and no match is found, total fragment is comment *.
3.  IdentifiedComment is known with index from 'Repetitive Strain Injuries'.
        Insert ''de definitie van RSI'' on top.
        No cleaning necessary.
4.  No topic fronting, because topic already placed in front segment.
```