Knowledge-based query formulation in information retrieval

Summary

Knowledge-based Query Formulation in Information Retrieval

Our research resides, in the field of Information Retrieval. It aims at improving the effectiveness of *query-based document-search processes*. These processes comprise four sub-processes, viz. (1) describing of documents (usually indexing), (2) query formulating, (3) matching, and (4) inspecting. In a matching process a query is compared to an index list of a document collection, thereby calculating for each document a relevance value for the query. The query is a formal expression meant to indicate the information need of a user; the indexes are expressions meant to characterise the documents in the collection. Thus, a matching process yields a list of documents predicted to be relevant to the user's information need. After inspection, a new query may be formulated using the information gathered. The latter process is called relevance feedback. The effectiveness of query-based document-search processes is expressed by the notions recall and precision. An effective process has a high recall and a high precision, i.e., it identifies many of the relevant documents in the document collection and it identifies hardly any irrelevant documents.

Query-based document-search processes show an effectiveness (recall and precision) which is far from perfect. One possible cause for the poor performance is that the queries formulated are imperfect. This leads to our main research question, in which we restrict ourselves to initial queries (i.e., queries formulated in which no relevance feedback has been applied):

> *How can, in query-based document-search processes, the effectiveness of the formulation of an initial query be improved by utilising represented knowledge?*

The main research question is worked out in three research questions. Here we restrict ourselves to query-based document-search processes using full-text indexing and a Boolean query language. The first research question aims at studying the formulation of queries by human beings. The second question prompts the development of a representation language (Dipe-R) for domain knowledge that supports initial query-formulation processes. The third question aims at the

development of a knowledge-based technique that supports initial query-formulation processes. The latter technique incorporates a two-step approach (substantiated in a program named Dipe-D) in which first the concepts of an information need are determined, and then the actual query is formulated. Both Dipe-R and Dipe-D are used in an experiment to test Dipe-D's effectiveness. The main conclusion of our research is:

> *The effectiveness of query-based document-search processes, with full-text indexing and Boolean matching, can be improved by the two-step approach to initial query formulation of Dipe-R and Dipe-D.*

Chapter 1 introduces query-based document-search processes, by their concepts and performance measures. It also motivates the relevance of our research, indicating that the processes are important in daily life, that they are not highly effective, and that the latter finding can be partly explained by problems in query formulation.

Chapter 2 characterises basic notions that play a role in discussing document-search processes: communication, information, knowledge, and document. Although – or rather because – these are familiar notions, we believe it to be wise to characterise what *we* mean by them, for the sake of effective communication.

Chapter 3 reviews the literature on Information Retrieval, and formulates the above-mentioned research questions. The review comprises various matching models, methods of indexing, and query formulating. Focusing on query formulating, we find that all existing query languages have keys and most of them also have operators. Known difficulties in query formulation are:

1. that a user cannot express his information need in a proper way, since he cannot
   a. use the appropriate operators in a proper way,
   b. conceive the subjects of the information need,
   c. designate the subjects properly (i.e., in correspondence to the document descriptors), and
2. that a user does not know whether the subjects of the information need are present in the document collection, i.e., he does not know whether the document collection can satisfy the information need (this is called a conceptual gap).

It is known that queries can be improved by reformulation, with (1) domain information, and (2) relevance feedback. We expect that a good initial query will converge rather fast to a high effectiveness of the search process, either with or without query reformulation. We regard it as our challenge to improve the effectiveness of the initial query formulation process, by using represented domain knowledge. Hence we arrived at the above-mentioned research questions.

Chapter 4 treats the first research question, aiming at an improved understanding of query formulation: *How are queries formulated adequately, in a search task that serves the writing of a patent application document?* A theoretical analysis of query formulation yields the insight that two sub-processes occur (possibly intertwined): (1) developing an understanding of the knowledge lack, and (2) expressing the information need. A simulation of patent-search tasks yields the following three (tentative) conclusions: (1) the user's understanding of his knowledge lack (i.e., his information need) changes noticeably, i.e., the development of the understanding of the

knowledge lack plays an essential role, (2) concepts are identified by their characteristics (direct or by association), and (3) queries are formulated according to a fixed procedure.

Chapter 5 treats the second research question: *Is it possible to formulate a representation language for domain knowledge that effectively supports initial query-formulation processes?* Developing Dipe-R assesses the question. Dipe-R's general properties are:

1. the representation of thoughts,
2. a distinction between the storage form and presentation form of represented thoughts. The presentation form is cast in sentences of natural language. This supports the language variety needed in query formulation,
3. a distinction between the content of a represented thought and its source: each represented thought is accompanied by source information, existing of (a) an identifier of the person or document which is the source, (b) the moment in which the thought was obtained, and (c) the manner in which it was obtained (observation, communication, etc.). This distinction recognises the dynamic and subjective nature of thoughts (and of the way of expressing). It is deemed useful for finding a proper interpretation of the sentence, and for assessing the credibility of the thought it expresses.

More specific, Dipe-R has two types of represented thoughts. Both types represent binary relations between concepts. One type also includes quantities. By these two types of thoughts, type hierarchies as well as features of concepts can be represented. Moreover, relationships among concepts, designations, and words are represented. The latter is realised by recognising that designations and words, i.e., symbols, are also concepts. In the same manner as other concepts, the binary relations of the two thought types of Dipe-R thus can relate symbols to concepts (including symbols). Only when a represented thought containing a symbol concept is presented to a user, the symbol concept is treated differently from other concepts (an instance is shown). In addition, Dipe-R contains derivation rules and a (simple) derivation procedure. These allow for elementary derivations, such as the inheritance of features, and the use of transitivity. With this structure of Dipe-R, concepts can be identified by their hierarchical relationships and their features.

Chapter 6 treats the third research question: *Is it possible to formulate a knowledge-based technique that effectively supports initial query-formulation processes?* Developing Dipe-D, including a two-step approach based on the lessons of chapter 4, assesses the question. The first step serves the development of the information need, and the identification of concepts. It comprises an iterative loop of posing questions and obtaining answers with explanations. The basic concept-identification loop consults a knowledge representation expressed in Dipe-R, and has four steps: (1) specification of the information need (in an artificial language), (2) disambiguation (if necessary), (3) solution, i.e., identifying represented concepts that meet the specification, and (4) explanation. An extended loop repeats the basic loop several times, in order to create descriptions of information needs that cannot be expressed in single concepts of the knowledge representation consulted.

The language for specification consists of two main expression types, and a small collection of sub-expression types. The main expressions allow for starting a specification and extending another one, respectively. The sub-expressions specify concepts by type hierarchy, and by (other) relations to other concepts. By combining several sub-expressions, a variety of specifications can

be created. The specification language allows for both a broad description and a narrow description. By using the answers, the user may gather the knowledge needed to develop his information need and to create a proper expression thereof. The narrow descriptions of the specification language then may be used to express precisely an established information need, yielding only the concepts needed.

Dipe-D's second step serves the (automatic) transformation of the collection of (represented) concepts identified into a Boolean query. In its basic version the transformation literally uses the designations (including synonyms) of the concepts, and connects them by the disjunction operator. A variant enhances the recall, as follows: it finds alternative designations of the concept (synonyms and lexical variants), creates a sub-query with the individual words (and word variants, and word stems, excluding stopwords, i.e., words known to have little distinguishing power) of each designation, and combines the sub-queries into a query. A next variant further enhances the recall, by adding quasi-synonyms. Another variant enhances the precision: it reduces the ambiguity of a homonymous designation in a query, by including designations related to the intended meaning and excluding terms related to unintended meaning(s) of the homonymous designation. The transformation process described serves the basic concept-identification loop above. The extended loop is served by an extended transformation process: this is essentially identical, but is repeated for each time the basic loop is repeated, and subsequently the resulting (sub)queries are connected by conjunction operators into a single query.

Chapter 7 reports on the implementation of Dipe-D and on an example knowledge representation in Dipe-R. Both are used in the experiments of chapter 8 for testing Dipe-D. The implementation of Dipe-D is mainly done in the programming environment Delphi 2.0 (using the MsWindows operating system). The derivation part is programmed in PROLOG. The program of Dipe-D has two main windows: one for each step. The example knowledge representation lies in the engineering field of refrigeration. It contains approximately 2,300 represented thoughts, including 600 represented concepts. Despite the finding that many thoughts cannot be expressed in Dipe-R, we found it possible to express a fair amount of domain knowledge in Dipe-R.

Chapter 8 reports on experiments with Dipe-D. Each of Dipe-D's two steps is tested in isolation. The concept-identification step is tested by having six test persons each completing three search tasks. In each task, the test person had to identify concepts using Dipe-D with our example representation in Dipe-R, starting from a written search task. We draw three conclusions:

1. Dipe-D supports the user in developing an information need from a given description by features in natural language into a collection of concepts satisfying the description; this yields a concept precision of 100%, i.e., no irrelevant concepts are identified.
2. Dipe-D supports the user in developing an information need from a description by features in natural language into a collection of concepts satisfying the description with a concept recall of some 70%, i.e., less than 100%. In other words, not all relevant concepts are identified.
3. The causes for the observed low concept recall are (1) difficulties in translating the expression of types and features in the tasks into the types and features in the knowledge representation, and (2) insufficient expressive power of Dipe-D's specification language.

The transformation step was tested by comparing transformation by hand to automated transformation by Dipe-D. Eight test persons each completed three query-formulation tasks, in which

tasks the terms and their variants were given to the test persons. Dipe-D did the same tasks. Next, the queries were matched by a Boolean search system, having a document collection of patent abstracts. We draw the conclusion that the queries created by the transformation step of Dipe-D are on average as effective as those created manually by test persons familiar with Boolean document-search processes. Each of the four conclusions in chapter 8 is only tentative, since the scale of the experiments is small. Statistical significance may be obtained in further experiments, as well as proof that the transformation step by Dipe-D outperforms human beings.

Chapter 9 summarises the three research questions, their approaches and the results and conclusions. It ends with the main conclusion given above.

Samenvatting

Kennisgebaseerd formuleren van zoekvragen in Information Retrieval

Het in dit proefschrift beschreven onderzoek is gelegen in het gebied van *Information Retrieval*. Het onderzoek richt zich op het verbeteren van de effectiviteit van *zoekvraaggebaseerde documentenzoekprocessen*. Dergelijke processen omvatten vier deelprocessen, te weten (1) het beschrijven van documenten (meestal indexeren), (2) het formuleren van een zoekvraag, (3) het vergelijken, en (4) het inspecteren. In een vergelijkingsproces wordt een zoekvraag vergeleken met een indexlijst van een documentenverzameling, waarbij voor elk document een relevantiewaarde wordt berekend bij de zoekvraag. De zoekvraag is een formele uitdrukking, die bedoeld is om de informatiebehoefte van een gebruiker aan te duiden; de indexen zijn uitdrukkingen, die bedoeld zijn om de documenten in de verzameling te karakteriseren. Een vergelijkingsproces levert aldus een lijst van documenten die relevant worden geacht voor de informatiebehoefte van de gebruiker. Na het inspecteren kan een nieuwe zoekvraag worden geformuleerd met behulp van de verkregen informatie. Het laatstgenoemde proces wordt relevantieterugkoppeling genoemd. De effectiviteit van zoekvraaggebaseerde documentenzoekprocessen wordt uitgedrukt in *recall* en *precision*. Een effectief proces heeft een hoge recall en een hoge precision, d.w.z., het identificeert bijna al de relevante documenten in de documentenverzameling, en het identificeert nauwelijks irrelevante documenten.

De effectiviteit (recall en precision) van zoekvraaggebaseerde documentenzoekprocessen is verre van perfect. Een mogelijke oorzaak voor de lage effectiviteit is dat de geformuleerde zoekvragen niet goed zijn. Dit brengt ons op onze hoofdonderzoeksvraag, waarin we onszelf tot initiële zoekvragen beperken (d.w.z. zoekvragen die zijn geformuleerd zonder dat relevantieterugkoppeling is uitgevoerd):

Hoe kan, in zoekvraaggebaseerde documentenzoekprocessen, de effectiviteit van het formuleren van een initiële zoekvraag worden verbeterd door het benutten van gerepresenteerde kennis?

De hoofdonderzoeksvraag wordt in drie onderzoeksvragen uitgewerkt. Hierbij beperken we ons tot zoekvraaggebaseerde documentenzoekprocessen met full-text indexering en een Boole'se zoekvraagtaal. De eerste vraag richt zich op het bestuderen van het formuleren van zoekvragen door mensen. De tweede vraag leidt tot het ontwikkelen van een representatietaal (Dipe-R) voor domeinkennis die formuleringsprocessen van initiële zoekvragen ondersteunt. De derde vraag houdt zich bezig met het ontwikkelen van een kennisgebaseerde techniek die formuleringsprocessen van initiële zoekvragen ondersteunt. Laatstgenoemde techniek bevat een tweestapsbenadering (die belichaamd wordt in een programma met de naam Dipe-D). Deze benadering stelt eerst de concepten uit een informatiebehoefte vast, en formuleert vervolgens de feitelijke zoekvraag. Zowel Dipe-R als Dipe-D zijn gebruikt in een experiment voor het testen van de effectiviteit van Dipe-D. De hoofdconclusie van ons onderzoek luidt:

> *De effectiviteit van zoekvraaggebaseerde documentenzoekprocessen, met full-text indexering en een Boole's vergelijkingsproces, kan worden verbeterd door middel van de tweestapsbenadering voor het formuleren van initiële zoekvragen van Dipe-R en Dipe-D.*

Hoofdstuk 1 introduceert zoekvraaggebaseerde documentenzoekprocessen aan de hand van relevante begrippen en prestatiemaatstaven. Het geeft ook een motivering van ons onderzoek, waarbij aangegeven wordt dat de processen belangrijk zijn in het dagelijks leven, dat ze niet erg effectief zijn, en dat deze laatste bevinding ten dele kan worden verklaard uit de moeilijkheden bij het formuleren van zoekvragen.

Hoofdstuk 2 kenmerkt basisbegrippen die een rol spelen bij het bespreken van documentenzoekprocessen: communicatie, informatie, kennis, en document. Alhoewel – of liever omdat – dit bekende begrippen zijn, achten wij het raadzaam om aan te geven wat *wij* er mee bedoelen, ter bevordering van een effectieve communicatie.

Hoofdstuk 3 bespreekt literatuur over Information Retrieval, en formuleert de hogergenoemde onderzoeksvragen. De bespreking omvat vergelijkingsmodellen, indexeringsmethoden, en het formuleren van zoekvragen. Onze aandacht is in eerste instantie gevestigd op het formuleren van zoekvragen. Het is bekend dat alle bestaande zoekvraagtalen sleutels hebben en de meeste bovendien operatoren. Twee bekende moeilijkheden bij het formuleren van zoekvragen zijn:

1. dat een gebruiker zijn informatiebehoefte niet op de juiste manier kan uitdrukken, aangezien hij niet a. de juiste operatoren op de juiste manier kan gebruiken, b. de onderwerpen van de informatiebehoefte kan bedenken, c. de onderwerpen juist kan aanduiden (d.w.z. in overeenstemming met de documentbeschrijvingen), en
2. dat een gebruiker niet weet of de onderwerpen van de informatiebehoefte aanwezig zijn in de documentenverzameling, oftewel hij weet niet of de documentenverzameling de informatiebehoefte kan vervullen (dit wordt een conceptuele kloof genoemd).

Het is bekend dat zoekvragen kunnen worden verbeterd door herformulering, met (1) domeininformatie, en (2) relevantieterugkoppeling. Wij verwachten dat een goede initiële zoekvraag tamelijk snel naar een hoge effectiviteit van het zoekproces zal convergeren, ongeacht of een herformulering van de zoekvraag plaatsvindt. We beschouwen het als een uitdaging om de effectiviteit van het formuleringsproces van initiële zoekvragen te verbeteren, door gebruik te

maken van gerepresenteerde domeinkennis. De hierboven genoemde onderzoeksvragen zijn hierop gebaseerd.

Hoofdstuk 4 behandelt de eerste onderzoeksvraag, die gericht is op een verbeterd begrip van het formuleren van zoekvragen: *Hoe worden zoekvragen adequaat geformuleerd, in een zoektaak die het schrijven van een octrooiaanvraag dient?* Een theoretische analyse van het formuleren van zoekvragen leidt tot het inzicht dat er twee (mogelijk verweven) deelprocessen optreden: (1) het ontwikkelen van het begrip van het kennistekort (de informatiebehoefte), en (2) het uitdrukken van de informatiebehoefte. Een simulatie van zoektaken naar octrooien leidt tot de volgende drie (voorlopige) conclusies: (1) het begrip van de gebruiker in het eigen kennistekort (zijn informatiebehoefte) wijzigt aanzienlijk, d.w.z. de ontwikkeling van het begrip van het kennistekort speelt een wezenlijke rol, (2) concepten worden geïdentificeerd aan de hand van hun kenmerken (rechtstreeks of via associatie), en (3) zoekvragen worden volgens een vaste procedure geformuleerd.

Hoofdstuk 5 behandelt de tweede onderzoeksvraag: *Is het mogelijk om een representatietaal voor domeinkennis te formuleren die op effectieve wijze initiële formuleringsprocessen ondersteunt?* De aanpak van deze vraag leidt tot de ontwikkeling van Dipe-R. De algemene kenmerken van Dipe-R zijn:

1. de representatie van gedachten,
2. het onderscheiden van de opslagvorm en presentatievorm van gerepresenteerde gedachten. De presentatievorm bestaat uit zinnen in natuurlijke taal. Dit ondersteunt de taalverscheidenheid die nodig is bij het formuleren van zoekvragen,
3. het onderscheiden van de inhoud van een gerepresenteerde gedachte en zijn bron: elke gerepresenteerde gedachte wordt vergezeld van broninformatie, die bestaat uit (a) een aanduiding van de persoon of het document dat de bron is, (b) het moment van verwerving van de gedachte, en (c) de wijze van verwerving (waarneming, communicatie, etc.). Dit onderscheid onderkent de tijdsafhankelijke en subjectieve aard van gedachten (en van de wijze van uitdrukken). Het wordt van belang geacht voor het vinden van een juiste interpretatie van de zin, en voor het inschatten van de geloofwaardigheid van de gedachte die erdoor wordt uitgedrukt.

Meer in het bijzonder heeft Dipe-R twee typen gerepresenteerde gedachten. Beide typen representeren relaties tussen concepten. Een type bevat ook hoeveelheden. Met deze twee typen gedachten is het mogelijk om zowel typehiërarchieën als kenmerken van concepten te representeren. Bovendien worden relaties tussen concepten, aanduidingen, en woorden gerepresenteerd. Het laatste wordt verwezenlijkt door te onderkennen dat aanduidingen en woorden, d.w.z., symbolen, ook concepten zijn. Op dezelfde wijze als andere concepten kunnen de binaire relaties uit de twee gedachtetypen in Dipe-R dus symbolen relateren aan concepten (inclusief symbolen). Pas wanneer een gerepresenteerde gedachte die een symboolconcept bevat aan de gebruiker wordt gepresenteerd wordt het symboolconcept anders behandeld dan andere concepten (er wordt een exemplaar getoond). In aanvulling op het voorgaande bevat Dipe-R afleidingregels en een (eenvoudige) afleidingsprocedure. Deze maken eenvoudige afleidingen mogelijk, zoals het overerven van kenmerken, en het gebruiken van transitiviteit. Met deze structuur van Dipe-R kunnen concepten worden geïdentificeerd aan de hand van hun hiërarchische relaties en hun kenmerken.

Hoofdstuk 6 behandelt de derde onderzoeksvraag: *Is het mogelijk om een kennisgebaseerde techniek te formuleren die op effectieve wijze initiële formuleringsprocessen van zoekvragen ondersteunt?* De ontwikkeling van Dipe-D behandelt deze vraag, met een tweestapsbenadering die gebaseerd is op de lessen uit hoofdstuk 4. De eerste stap dient voor het ontwikkelen van de informatiebehoefte en het identificeren van concepten. Het omvat een iteratieve lus van vragen stellen en antwoorden krijgen met uitleg. De elementaire concept-identificatielus raadpleegt een kennisrepresentatie uitgedrukt in Dipe-R en bevat vier stappen: (1) specificatie van de informatiebehoefte (in een kunstmatige taal), (2) desambiguering (indien nodig), (3) het oplossingsproces, d.w.z. identificeren van gerepresenteerde concepten die voldoen aan de specificatie, en (4) uitleg. Een uitgebreide lus herhaalt de elementaire lus verscheidene keren, teneinde beschrijvingen van informatiebehoeften te creëren die niet kunnen worden uitgedrukt in enkelvoudige concepten uit de geraadpleegde kennisrepresentatie.

De specificatietaal bestaat uit twee hoofdexpressietypen en een kleine verzameling subexpressietypen. De hoofdexpressies maken het mogelijk respectievelijk een specificatie te beginnen en uit te breiden. De subexpressies specificeren concepten aan de hand van een typehiërarchie en door middel van (andere) relaties tot andere concepten. Door meerdere subexpressies te combineren kan een verscheidenheid van specificaties worden gecreëerd. De specificatietaal maakt zowel een ruime als een nauwe beschrijving mogelijk. Door de antwoorden te benutten, kan de gebruiker de kennis vergaren die nodig is om zijn informatiebehoefte te ontwikkelen en om een juiste uitdrukking daarvan te creëren. De nauwe beschrijvingen van de specificatietaal kunnen dan worden gebruikt om een gerealiseerde informatiebehoefte zodanig precies te beschrijven, dat de beschrijving slechts de benodigde concepten oplevert.

De tweede stap van Dipe-D dient voor het (automatisch) transformeren van de verzameling geïdentificeerde (gerepresenteerde) concepten tot een Boole'se zoekvraag. In de basisuitvoering gebruikt de transformatie letterlijk de aanduidingen (inclusief synoniemen) van de concepten en verbindt ze door de disjunctie operator. Een variant bevordert de recall, op de volgende wijze: hij vindt andere aanduidingen van het concept (synoniemen en lexicale varianten), creëert een subzoekvraag met de losse woorden (en woordvarianten, en woordstammen, met uitzondering van stopwoorden, d.w.z. woorden die bekend zijn om hun geringe onderscheidend vermogen) van elke aanduiding, en combineert de subzoekvragen tot een zoekvraag. Een volgende variant bevordert de recall verder, door quasi-synoniemen toe te voegen. Nog een andere variant bevordert de precision: hij verkleint de ambiguïteit van een homonieme aanduiding in een zoekvraag, door in de zoekvraag aanduidingen te noemen die zijn gerelateerd aan de bedoelde betekenis, en aanduidingen uit te sluiten die zijn gerelateerd aan de onbedoelde betekenis(sen) van de homonieme aanduiding. Het beschreven transformatieproces dient voor de bovengenoemde elementaire concept-identificatielus. Voor de uitgebreide lus is een uitgebreid transformatieproces: dit is in hoofdzaak identiek, maar wordt elke keer herhaald dat de elementaire lus wordt herhaald. Vervolgens worden de resulterende (deel)zoekvragen door conjunctie operatoren verbonden tot een enkele zoekvraag.

Hoofdstuk 7 rapporteert de implementatie van Dipe-D en een proefkennisrepresentatie in Dipe-R. Beide worden gebruikt in de experimenten uit hoofdstuk 8 voor het testen van Dipe-D. De implementatie van Dipe-D gebeurt in hoofdzaak in de programmeeromgeving Delphi 2.0 (onder het MsWindows besturingssysteem). Het afleidingsgedeelte is in PROLOG geprogrammeerd.

Het programma van Dipe-D heeft twee hoofdschermen: één voor elke stap. De proefkennisrepresentatie ligt op het technische vakgebied koudetechniek. Hij bevat ongeveer 2.300 gerepresenteerde gedachten, daarbij zijn 600 gerepresenteerde concepten inbegrepen. Ondanks de bevinding dat menig gedachte niet in Dipe-R kan worden uitgedrukt, hebben wij ervaren dat het mogelijk is om een flinke hoeveelheid domeinkennis in Dipe-R uit te drukken.

Hoofdstuk 8 rapporteert experimenten met Dipe-D. Elk van de twee stappen uit Dipe-D is afzonderlijk getest. De concept-identificatiestap is getest door zes personen ieder drie zoekopdrachten te laten uitvoeren. In elke taak moet de testpersoon concepten identificeren met Dipe-D en onze proefrepresentatie in Dipe-R. Er wordt uitgegaan van een geschreven zoekopdracht. We trekken drie conclusies:

1. Dipe-D ondersteunt de gebruiker bij het ontwikkelen van een informatiebehoefte uit een gegeven beschrijving in natuurlijke taal op grond van kenmerken tot een verzameling concepten die elke voldoen aan de beschrijving; dit resulteert in een concept precision van 100%, d.w.z. dat er geen irrelevante concepten worden geïdentificeerd.
2. Dipe-D ondersteunt de gebruiker in het ontwikkelen van een informatiebehoefte uit een gegeven beschrijving in natuurlijke taal op grond van kenmerken tot een verzameling concepten die voldoen aan de beschrijving met een concept recall van ongeveer 70%, d.w.z. minder dan 100%. Anders gezegd, niet alle relevante concepten worden geïdentificeerd.
3. De oorzaken van de waargenomen lage concept recall zijn (1) moeilijkheden bij het vertalen van de uitdrukkingswijze van typen en kenmerken in de zoekopdrachten naar de uitdrukkingswijze van typen en kenmerken in de kennisrepresentatie, en (2) onvoldoende uitdrukkingskracht van de specificatietaal uit Dipe-D.

De transformatiestap werd getest door de handmatige transformatie te vergelijken met de geautomatiseerde transformatie door Dipe-D. Acht proefpersonen voeren elk drie formuleringsopdrachten van zoekvragen uit. In de opdrachten zijn de termen en hun varianten aan de proefpersonen gegeven. Dipe-D voert dezelfde opdrachten uit. Vervolgens zijn de zoekvragen verwerkt door een Boole's zoeksysteem met een documentenverzameling uit octrooi-uittreksels. We concluderen dat de zoekvragen die door de transformatiestap van Dipe-D zijn gecreëerd gemiddeld net zo effectief zijn als de zoekvragen die handmatig zijn gecreëerd door proefpersonen welke bekend zijn met Boolese documentenzoekprocessen. Elk van de vier conclusies in hoofdstuk 8 is slechts voorlopig, aangezien de omvang van de experimenten gering is. Aanvullende experimenten kunnen statistische significantie opleveren, alsmede aantonen dat de tweede stap van Dipe-D het van de mens wint.

Hoofdstuk 9 resumeert de drie onderzoeksvragen, de aanpakken ervan, en de resultaten en conclusies. Het eindigt met de hierboven vermelde hoofdconclusie.