Creation and Use of Language Resources in a Question-Answering eHealth System

Ulrich Andersen[1], Anna Braasch[2], Lina Henriksen[2], Csaba Huszka[3], Anders Johannsen[2], Lars Kayser[4], Bente Maegaard[2], Ole Norgaard[4], Stefan Schulz[3,5], Jürgen Wedekind[2]

[1]Sorano Ltd.
[2]University of Copenhagen, Faculty of the Humanities, Centre for Language Technology
[3]Freiburg University Medical Center, Institute of Medical Biometry and Medical Informatics
[4]University of Copenhagen, Faculty of Health Sciences
[5]Medical University of Graz, Institute for Medical Informatics, Statistics and Documentation

Abstract

ESICT (Experience-oriented Sharing of health knowledge via Information and Communication Technology) is an ongoing research project funded by the Danish Council for Strategic Research. It aims at developing a health/disease related information system based on information technology, language technology, and formalized medical knowledge. The formalized medical knowledge consists partly of the terminology database SNOMED CT and partly of authorized medical texts on the domain. The system will allow users to ask questions in Danish and will provide natural language answers. Currently, the project is pursuing three basically different methods for question answering, and they are all described to some extent in this paper. A system prototype will handle questions related to diabetes and heart diseases. This paper concentrates on the methods employed for question answering and the language resources that are utilized. Some resources were existing, such as SNOMED CT, others, such as a corpus of sample questions, have had to be created or constructed.

Keywords: question answering, eHealth, SNOMED CT, summarization, question generation

1. Introduction

Existing open-domain question-answering (QA) systems, whether based on statistics or on ontologies, have various limitations with respect to domain coverage, the size and usability of the underlying texts, the resources available for maintenance, etc. Currently available eHealth systems for Danish are not based on QA technologies, but employ traditional (Google-like) searches. They mainly retrieve pre-existing texts (not customized answers) that provide general insights into the subject, but are not necessarily suitable in the individual citizen's case.

The ESICT system will be a hybrid QA system. It is based on different interacting methods of answer retrieval, uses structured, semi- or unstructured sources, will allow users to ask questions in Danish, and will provide natural language answers. This contribution focuses on question corpora, resources, and methods applied for generation of answers within the field of diabetes mellitus.

Three approaches to question processing and answer generation are proposed. Each approach is based on different written language resources (LRs): question corpora, medical texts and terminological/ontological resources. In approach ***A***, a deep semantic analysis of user questions forms the basis of queries in a medical term base/ontology. Approach ***B*** relies on query-focused multi-document summarization. Finally, in approach ***C***, we generate potential users' questions that have viable answers in a medical document collection.

2. Collection of Real-Life Questions

We identified the scope of health care users' information needs by collecting real-life questions from three different sources: an online diabetes discussion forum (henceforth OLF), people with diabetes attending an outpatient clinic at a Danish hospital (Wizard of Oz (Kelley, 1983) (WoZ) sessions), and a workshop with health informatics students. Such a corpus is a prerequisite for defining the required system coverage.

The OLF is provided by the Danish Diabetes Association and runs on the organisation's website. More than 1,700 people are registered as users. Initially, we identified 1,123 threads starting with a question; in total 263 questions were collected from this source. For the WoZ sessions, we recruited eight outpatients at a specialized diabetes clinic at Bispebjerg Hospital, Denmark (one type 1 and seven type 2). Each participant received a list of seven scenarios where a diabetic will typically need information and advice (e.g., when travelling abroad/being sick). In order to emulate practical use of the QA system, users asked questions from their own computer. The questions were answered promptly by a medical doctor (the wizard). To simulate a computerized question-answer process, the doctor's involvement was only revealed to the participant after the session had ended. The WoZ sessions produced 195 questions. Moreover, we held a workshop with 32 health informatics students. They were asked to write down, in 30 minutes, as many – and preferably concise – questions about diabetes as they could think of.

The corpus revealed 6 main topics including *Molecular and biomedical facts*, *Epidemiology*, *Interventions*, and *Diagnostics*. Each main topic is further divided in up to 6 subtopics. Around 60% of questions from WoZ and OLF fell within the main topic *Interventions* and mainly concerned the subtopic *Behavioral intervention* (diet, exercise, and life style issues). The remaining questions from these sources were distributed fairly evenly over all other topics. Questions generated by health informatics students also reflected life-style issues, but only for approximately 33% of the questions; 25% of the questions fell within *Molecular and biomedical facts* and another 25% in the *Epidemiology* group. The significant difference in students' and patients' question topic distribution is probably caused by the students' neutral perspective and patients' personal involvement.

3. Knowledge-based Question Answering – Approach *A*

Approach ***A*** draws on SNOMED CT[1], a comprehensive multilingual clinical terminology collection covering a wide range of medical specialities. The basic idea is to transform natural language questions into SQL queries and to produce natural language answers based on SNOMED's output. The idea of using SNOMED CT as an important resource for answer material identification is completely novel.

3.1 Toward the SQL Query

The conversion of the questions into SQL queries involves syntactic parsing and (quasi) semantic interpretation of the questions. For syntactic parsing we use the second-order non-projective model of MSTParser (McDonald & Pereira, 2006) (trained with 5-best MIRA) with Google-tagset (Petrov et al., 2012). As POS-tagger we use the SVMTool. We demonstrate the various steps of question processing with a typical example. For the question (1)

(1) Er diabetes arveligt? (Is diabetes hereditary?)

the parser produces the output in (2).

(2) ROOT(ROOT, Er) subj(Er, diabetes)
pred(Er, arvelig) pnct(Er,?)
(ROOT(ROOT, Is) subj(Is, diabetes)
pred(Is, hereditary) pnct(Is,?))

The graphical representation of (2) is given in figure 1.

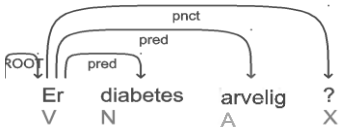


Figure 1: the parsing result for (1).

In the second step, we construct from the dependency structure of the question a (semantic) representation that is interpretable in SNOMED CT. For (2) this representation is given in (3).

(3) 'diabetes' AND 'arvelig'
('diabetes' AND 'hereditary')

These representations serve as input to the SQL query generation. For several types of simple factoid questions, SQL templates have been developed allowing mapping of semantic representations into SQL queries. Complex factoid questions are not processed in approach ***A***.

Table 1 shows the SQL query for (3). This query is one single combined query retrieving the intersection of concepts represented by terms containing the string 'diabetes', together with all their taxonomic descendants AND concepts represented by terms containing the string 'arvelig' together with all their taxonomic descendants. For this particular question, SNOMED CT answer material includes retrieved terms as, for example, 'hereditær nefrogen diabetes insipidus' (hereditary nephrogenic diabetes insipidus).

SQL query
SELECT * from concepts where conceptID in ((SELECT * FROM concepts WHERE (TERM ('diabetes') > 0)) UNION ALL (SELECT DISTINCT conceptID from concepts WHERE parentID in (SELECT * FROM concepts WHERE (TERM ('diabetes') >0)))) INTERSECT (SELECT * FROM concepts WHERE (TERM ('arvelig') > 0)) UNION ALL (SELECT DISTINCT conceptID from concepts WHERE parentID in (SELECT * FROM concepts WHERE (TERM ('arvelig ') >0)))))
SNOMED output (here shown in English)
Diabetes mellitus autosomal dominant type II Glycogenosis with glucoaminophosphaturia Haemochromatosis Hemochromatosis Hereditary benign acanthosis nigricans with insulin resistance Hereditary nephrogenic diabetes insipidus Maturity onset diabetes mellitus in young

Table 1: SQL query on a relational table 'snomed', which contains the transitive closure over all isA (child/parent) relations of SNOMED CT. The function "TERM" looks up for exact string matches in a related table with terms ('descriptions') and returns the number of matches.

3.2 Toward Natural Language Answers

The question (1) is a polar question (yes/no). Since the retrieval is not empty, concepts describing diabetes with

[1] www.ihtsdo.org/snomed-ct
SNOMED Clinical User Guide, International Health Terminology Standards Development Organization, January 2010, International Release.

hereditary conditions do exist. Therefore at least part of the answer is 'yes', though questions where a simple 'yes' or 'no' will be satisfactory are clearly exceptions. Generally, an affirmative or negative answer must be followed by a more elaborate explanation similar to the answers to factoid questions.

As regards answer material, SNOMED CT output will as shown in table 1 contain terms, and may also contain relations and hierarchy names. However, SNOMED CT retrievals are more or less illegible for the user in this format. Therefore natural language answers must be generated. This task is currently in progress and comprises, for example, ranking of answer candidates, transformation of SNOMED CT's terminology into general language expressions, and linguistic structuring of the answer.

Answer generation will be based on pre-defined question-answer pattern pairs. Our preliminary studies show that a considerable number of the questions in the corpus and their respective answers can be captured through a limited number of question-answer pattern pairs.

The linguistic generation of the example polar question will be based on answer patterns like 'Yes, *X* is *Y*' or 'No, *X* is not *Y*'. In this example an answer could be 'Yes, diabetes is hereditary in the following examples: hereditary nephrogenic diabetes insipidus, …'.

The simple factoid question (4), for example, requires a more elaborate answer.

(4) Hvilke symptomer er der på diabetes type 2?
(What symptoms are connected with diabetes 2?)

Below we sketch the steps toward identification of the appropriate answer pattern. For (4) the parser produces the analysis in (5).

(5) subj(er,Hvilke) rel(Hvilke,symptomer)
ROOT(ROOT,er) expl(er,der) pred(er,på)
nobj(på,diabetes) nobj(diabetes,type) nobj(type,2)
pnct(er,?)

The graphical representation of (5) is depicted in figure 2.

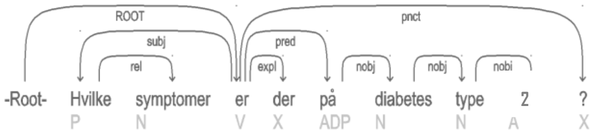


Figure 2: parsing result of question (4).

The SNOMED CT interpretable representation of (5) is (6).

(6) ASSOCIATED_WITH(finding, 'diabetes type 2')

SNOMED's output for the SQL query (6) and the answer pattern (7)

(7) [finding](subj) 'er symptomer på' [disorder](nobj)

associated with questions of the form (4) will allow generation of the answer (8).

(8) [search results] er symptomer på diabetes type 2
([search results] are connected with diabetes type 2)

3.3 SNOMED CT Coverage

We investigated the coverage of SNOMED CT in relation to user questions on a subset of randomly selected questions (138) from the collected corpus. The investigation included retrieval of question keywords and the relations appearing between the concepts of the question keywords. The investigation revealed that the vast majority of medical as well as medically related concepts, appearing in the question corpus as a whole, exists in SNOMED CT. However, it also revealed that a number of question concepts, especially those appearing in the lifestyle topic questions, are not covered by SNOMED CT.

4. Query-based Summarization – Approach *B*

Some questions have short and precise answers, requiring perhaps a sentence or two to be addressed in a satisfactory manner. Other questions ask for explanations. 'What', 'How', or 'Why' often are the first word in such questions. Some examples taken from our corpus are: 'What is the difference between diabetes type 1 and 2?', 'How many meals a day should I take?', and 'Why do I get swollen feet when I sit still for long period of time, for instance at a dinner party?'. A good answer will typically be paragraph-sized, with text narrowly focused on the user's information need.

Approach ***B*** addresses questions that have no simple factual answers. Complex answers are generated by a document summarization algorithm that is query-focused and compiles a summary on the basis of multiple documents. Below, we describe the algorithm along with modifications that make it more suitable for the ESICT project. Then we consider a problem facing question answering as summarization: the possible mismatch between very specific and detailed questions ('Can I eat two apples and a large pear for breakfast without upsetting my blood sugar?'), and answers of a more generic variety. The final section discusses requirements on the answer section of the corpus that are relevant for the evaluation of approach ***B***.

4.1 Content Selection and Text Coherence

Multi-document summarization is an open research topic, and one difficult problem is diversification. When doing extraction from a single document, the most important sentences are identified and glued together to form a summary. Clearly, this will not work unaltered in case of multiple documents since the most important sentences across the documents are likely to be quite similar to each other. A summary composed in this fashion would thus be highly redundant.

One common way of avoiding similar sentences in the output is to choose them one at a time and only include a sentence if it is both relevant and not redundant with respect to the sentences already chosen. This is the widely used Maximum Margin Relevance (MMR) approach. Our concept selection algorithm is based on a different strategy: concept coverage maximization, where concepts are meaning units extracted from a sentence. A concept, say the fact that eating fruits causes your blood sugar to rise, is associated with a value saying how important it is. Concept coverage maximization seeks to extract the set of sentences that maximizes the value of the included concepts and fits within a given number of words, the summary length. As there is no sense in stating the same facts more than once, each concept only counts towards the score once regardless of how many times it appears. It is precisely calculating the score on the basis of unique concepts that puts a penalty on similar sentences, simply because a set of sentences with redundant content will have the opportunity to express fewer unique concepts when summary length is constrained. Additionally, this concept coverage maximization method recovers the globally optimal solution, whereas MMR greedy selection procedure comes with no such guarantee.

In the framework of concept coverage maximization, the key to obtaining good performance is defining viable concepts and assigning values to them in a sensible manner. The mapping from sentences to concepts can be as simple as bigrams, and as complicated as genuine semantic relations. In our initial experiments, we have used grammatical relations derived from dependency triplets to build concepts, with moderate success. Going forward, we plan to integrate information from medical ontologies for assigning weights to the concepts.

State-of-the-art summarization systems rely on ROUGE scores for evaluation. Typically, bigram overlap with a set of human produced summaries (ROUGE-2) is the metric of choice for comparison. While evaluation at the Document Understanding Conference (DUC), now superseded by Text Analysis Conference (TAC), includes assessment of text quality by human judges, ROUGE remains the most important score. Perhaps as a consequence, systems handle content selection extremely well, but produce sub-optimal text: incoherent, unfocused, and generally hard to read. Though a few systems attempt to do things like order sentences and fix dangling pronouns, the text quality component seems to be added almost as an after-thought. However, this late in the process, it is impossible to choose alternatives that result in more coherent summaries. Currently, we investigate a new approach where the content selection and text realization steps are collapsed. We model summarization as a joint optimization problem in the dual decomposition framework. One objective is concept coverage maximization, which rewards the selection of salient content. The other objective models local coherence between sentences. It assigns the largest score to a summary where focus is not lost between adjacent sentences – a summary that "keeps talking" about related things. A parameter adjusts the split between the two objectives. By jointly optimizing for content and linguistic quality, we expect to deliver summaries that read more fluently than those predicted by the select-and-revise model.

4.2 Specific Questions – Generic Answers

If our knowledge source is a fixed set of documents, however large, then there are limits on the answer capability of the system; indeed, it's only possible to deal with questions that have an answer in the document collection. Generally speaking, as the complexity of the questions grows and the questions become more intimately tied to the users' identities, the probability of finding an answer in the corpus drops.

How likely are we to be short of an answer when considering the questions of the ESICT corpus? Notably, there is extensive use of self-reference ('jeg', 'mig'; in English: 'I', 'me') in the questions, which seems to suggest they are in some way specific to the user. However, this turns out not to be the case; although many of the questions in the corpus contain instances of the personal pronoun 'jeg', the 'jeg' is usually best understood as referring in a generic way to someone who has diabetes. This, at least, is what is going on in the questions from the WoZ sessions and the workshop since they are all one or two sentences long and have no backstory.

The use of 'jeg' to refer to a sort of generic person contrasts with the situation in a clinic where 'jeg' clearly refers to the patient and where, additionally, the doctor has immediate access to the medical history of the patient and so is able to provide advice tailored specifically to the patient's situation. The user, we assume, is well aware of how much (or little) background information he has provided to the system. In most cases, it is very little. So even if the question says 'jeg', there probably is no expectation that the answer would be only valid for that person. This of course has implications for the kind of answers that should be generated by the system. Specifically, a good answer must include an explanation as the user will need a bit of understanding to adapt the answer to his situation.

Some of the questions address very specific concerns (the following are glossed examples from the corpus): 'What is best to eat, apples or pears?' – 'Can I have eggs?' – 'Is honey okay?'. Or even more specific: 'What drink to prefer for rice pudding: christmas brew or cranberry juice?'. Questions of this type are challenging for two reasons. First, it seems unlikely that there would be specific answers to such questions anywhere in our text sources. After all, there is no end to the variety of the things that people enjoy eating. As before, though, it may actually be preferable to give a generic answer – one that explains the principles. That is, the user will have learned that the sugar content is the key information needed to make the decision.

Second, because words like 'honey' and 'eggs' are not part of the generic answer text, and may indeed not be present at all in the document collection, we need a way of mapping specific questions to generic answers. An ontology identifying 'honey' and 'eggs' as kind of 'food' gets us part of the way, and we are currently investigating the use of unsupervised topic models for this same task.

4.3 Answer Section of the Corpus

There are two principal modes of evaluation for a question-answering system: human and automated. In the human evaluation scenario, you feed a section of the question corpus labeled the test set to the system. When the answers arrive, you show them to a human and ask him to rate for quality and information content. In the automatic scenario, you compare the output of the system with a resource containing the "true" answer and measure the overlap: the higher, the better. Such a resource is called a gold corpus. The two kinds of evaluations have different strengths and weaknesses, and to a degree, they serve different purposes. Human evaluations are easy to interpret, but they are not easy to come by. Asking humans to read and rate summaries is time-consuming and presumably expensive – not something that you would want to do repeatedly. Nonetheless, while the system is being developed and new ideas tried out, there is a need for frequent evaluation, and this is where automatic evaluation becomes indispensable.

For the purposes of repeated evaluation, we will use ROUGE, which is the standard method of automatic evaluation for extractive multi-document summarization. A key requirement of ROUGE is a set of model summaries written by humans. To accommodate this, an answer in the ESICT corpus will be a set of texts composed by humans, intended to summarize the relevant information in the document collection.

5. Question Generation – Approach *C*

Approach ***C*** is based on question generation. The goal is to identify sentences of domain specific documents that can serve as answers to potential questions. Any identified sentence is then transformed into questions that can be answered by the sentence. All question/answer pairs thus extracted are stored in a database. For question answering, a user's question is identified in the question-answer database and the corresponding answer is returned.

The resources required are (i) a collection of reliable (authorized), informative documents on the domain, such as *Medicinhåndbogen* and *sundhed.dk*, (ii) a grammar to parse the documents, and (iii) a set of transformation rules that generate questions from the syntactic parse of useful answer sentences. Thus approach ***C*** does not rely on deep semantic analysis (as approach ***A***) but is not entirely shallow either (as approach ***B***).

As a simple illustration, consider the Danish sentence (9) contained in the document collection.

(9) Sukkersyge er en tilstand, hvor nedsat effekt eller produktion af hormonet insulin nedsætter muskler og organers evne til at optage sukkerstoffer fra blodet. (Diabetes is a condition where the efficacy or production of the hormone insulin decreases muscles and body's ability to absorb sugars from the blood.)

Obviously, this sentence is a useful answer candidate to a question like (10).

(10) Hvad er sukkersyge? (What is diabetes?)

To identify potentially useful answers in a collection of documents, a set of manually created syntactic patterns, henceforth called *triggers*, is matched against the syntactic parse of the documents. For the simple example above, the trigger is rather trivial:

(11) [SUBJ er en NOBJ, hvor …]
([SUBJ is a NOBJ where …])

where SUBJ is a question topic term (contained in a list of such terms).[2] Crucial is the fact that there is a relative clause that provides essential information on the NOBJ 'tilstand' (condition) and that the SUBJ is domain specific, i.e., a term like 'sukkersyge' (diabetes), 'insulin' (insulin), etc., but not, for example, 'der' (there) or some off-topic term. Such candidates have to be excluded from consideration a priori although they may also occur in the documents.

Schematically, the rule that transforms the syntactic structure of sentence (9) into the corresponding question has the form (12).

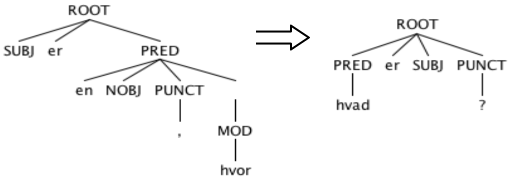
(12) [SUBJ er en NOBJ, hvor ...] → [Hvad er SUBJ?]
([SUBJ is a NOBJ where ...] → [What is SUBJ?])

To test the viability of the approach we implemented a prototype that generates factoid one-sentence questions.

[2] *Topic terms* are usually all the terms of a collection of documents that can be used as topic or focus of questions on the content of the documents. For our prototype, we obtained a list of on-topic terms from the Steno Diabetes Centre. However, in general the list of (domain-relevant) on-topic terms can be acquired automatically from a document collection by state-of-the-art statistical methods.

For the grammatical analysis we use a projective Danish dependency parser (Søgaard & Rishøj, 2010; McDonald & Pereira, 2006) and for question generation we use Tregex (a tree query language) and Tsurgeon (a tool for modifying trees) (Levy & Andrew, 2006). These tools match the syntactic descriptions of the triggers to syntactic analyses of the documents, apply the syntactic transformations to the document analyses, and output the resulting syntactic descriptions. For instance, the more precise Tregex description of the trigger (11) matches all analyses containing the left-hand side structure of (13). The Tsurgeon script converts the matching structures into structures of the form given by the right-hand side of (13).

(13)



The question generation approach seems particularly appropriate for factoid one-sentence questions. Apart from simple definitional questions as (10), it can also be used to produce more complex interrogative questions. From the sentence (14), for example, we can produce the ‘How‘ question in (15).

(14) Mild hypoglykæmi mærkes som svag svimmelhed, mens der i de alvorlige tilfælde er bevidsthedstab. (People with mild hypoglycemia experience dizziness while in the severe cases there is loss of consciousness.)

(15) Hvordan mærkes mild hypoglykæmi?
(How do people experience mild hypoglycemia?)

Example (16) illustrates that the question generation approach can also successfully create causative questions that are assumed to be inherently more difficult than other factoid questions.

(16) De almindeligste årsager til at få hypoglykæmi er for lidt kulhydrater i maden, for lang udskydelse af måltider, større fysisk aktivitet end sædvanligt eller for stor insulindosis. (The most common causes for getting hypoglycemia are not enough carbohydrates in the diet, too long deferral of meals, more physical activity than usual or excessive insulin dose.)

For (16) we can create the question in (17).

(17) Hvad er de almindeligste årsager til at få hypoglykæmi?
(What are the most common causes for getting hypoglycemia?)

Note that an equivalent ‘Why’ question can easily be produced by paraphrasing (see below): ‘Hvorfor får man hypoglykæmi?’ (Why does one get hypoglycemia?).

In the following we describe some obvious refinements and improvements of the basic setup that will be implemented in the next project phase.

5.1 Complex Answers

In many cases, complex answers can be created by extracting more than one sentence or by combining alternative answers.

Exploiting pronouns and definite noun phrases

Often, sentences that succeed an identified answer further elaborate that answer. This is done by pronouns and definite noun phrases that refer back to the identified answer. For instance, the following extract from a document

(18) I bugspytkirtlen dannes der hormoner, som bl.a. styrer kroppens sukkerbalance. Det drejer sig om glucagon og insulin. Insulinet gør, at det sukker, der er i blodet, nemmere optages i bl.a. muskel- og leverceller. (In the pancreas hormones are produced, which control, among others, the body’s sugar balance. These are glucagon and insulin. The insulin enables sugar in the blood to enter more easily, among others, muscle and liver cells.)

can, as a whole, be used to answer the question ‘Hvad dannes i bugspytkirtlen?’ (What is produced in the pancreas?) that is generated from the first sentence. This is because the succeeding sentences are connected to the first through a reference chain.

Exploiting multiple answers to the same question

In other cases, several answers that generate the same question can be combined to a more comprehensive answer (provided the answers are not uninformative variations of each other). Consider, for example, the two sentences in (19).

(19) Diabetes er en hyppigt forekommende livslang (kronisk) sygdom, og forekomsten er hastigt stigende så vel i Danmark som resten af verden. (Diabetes is a frequent lifelong (chronic) disease, and prevalence is rapidly increasing in Denmark as well as in the rest of the world.)
Diabetes er en alvorlig sygdom, da den medfører en betydelig risiko for udvikling af følgesygdomme i øjne, nyrer, nervebaner og blodkar. (Diabetes is a serious disease because it causes a significant risk for developing complications of the eyes, kidneys, nerves, and blood vessels.)

Both sentences generate the question ‘Hvad er diabetes?’ (What is diabetes?) and combined they provide a more comprehensive answer. In order to avoid that the combined sentences express uninformative variation, we will employ techniques developed in approach ***B*** to ensure that the combined sentences are maximally dissimilar to each other.

5.2 Clustering and Paraphrasing Questions

In many cases questions are generated that are meaning equivalent. For example, ‘Hvad er sukkersyge?’ and ‘Hvad er diabetes?’ convey in fact the same meaning and can thus be seen as belonging to the same cluster. Clustering can be used to improve the overall performance of the system, because it extends the number of answer sentences that can be combined to complex answers. This would, for example, enable subsequent processing to combine (9) and the two sentences in (19) to a more informative answer to (10). To further increase recall, existing questions can be paraphrased. This will usually improve the chances to provide an answer if the user enters a question that slightly varies from the questions contained in the database. Paraphrasing can be accomplished manually or semi-automatically, for example, by using medical ontologies, WordNets (the Danish DanNet), or phrase tables from existing machine translation systems.

5.3 Fuzzy Matching

Paraphrasing certainly improves the overall recall since it increases the number of questions that can be answered. However, in some cases a required paraphrase may not be included in the clusters, for example, because the user’s question differs only morphologically from a question in the database, as in ‘Hvad er symptomer på diabetes?’ and ‘Hvad er symptomerne på diabetes?’ (What are (the) symptoms of diabetes?). In these cases fuzzy matching (instead of exact matching) can be used to identify question candidates in the database. There exist various algorithms that can be used to compute the best matching database question. These range from relatively simple algorithms that compute the edit distance between two strings (e.g., Levenshtein distance computing algorithm) to more sophisticated ones that employ a translation model. To relativize the reliability of the answer, the system may return a ‘Did you mean [best matching database question]’-suggestion if no exact match was found.

6. Conclusion and Future Work

In this paper we have described the use of different types of language resources for the purpose of creating a QA system for ordinary citizens seeking information in the field of health. We have also briefly described those methods for question answering which we are pursuing. Even if they are not fully developed, we have demonstrated that they will all contribute to the performance of the QA system, i.e., the system will be truly hybrid. The next phase of the project will show how useful the different approaches are – how precisely will they answer questions, how many of the questions will be answered, etc.

7. Acknowledgements

The ESICT project is supported by the Danish Council for Strategic Research. All project participants are contributing to the work, and we thank those who are not authors of this paper for their contribution.

8. References

Heilman, M., Smith, N.A. (2010). Good question! Statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings*, Los Angeles, CA, pp. 609–617.

Kelley, J.F. (1983). An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of ACM SIG-CHI ’83 Human Factors in Computing Systems*, New York: ACM, pp. 193–196.

Levy, R., Andrew, G. (2006). Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, pp. 2231–2234.

McDonald, R., Pereira, F. (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, pp. 81–88.

Petrov, S., Dipanjan, D., McDonald, R. (2012). A universal part-of-speech tagset. To appear in *8th International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul.

Søgaard, A., Rishøj, C. (2010). Semi-supervised dependency parsing using generalized tri-training. In *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, pp. 1065–1073.