Learning Content Models for Semantic Search

ARD

Professional Advisor:
Michael Elhadad

Academic Advisor:
Menahem Adler

Team Members:
Eran Peer
Hila Shalom

Contents

Introduction 3
Vision 3
The Problem Domain 3
Stakeholders 5
Software Context 5
System Interfaces 10
Hardware Interfaces 10
Software Interfaces 11
Events 13
Functional Requirements 14
Non-functional requirements 15
Performance constraints 15
Platform constraints 15
SE Project constraints 16
Usage Scenarios 17
User Profiles – The Actors 17
Use-cases 18
Appendices 22

1 Introduction

1.1 Vision

The project goal is to allow the user to explore a repository of textual documents, and discover material of interest even when he is not familiar with the content of the repository.

Existing search engines provide powerful features to identify known documents by using a short description of their content (keywords, name of document). For example, a user wants to find information about a term that he knows, he enters the term's name in a search engine, and the search engine returns the address of the term's value at Wikipedia.

The taskwe address is what happens when the user does not know the name of the term he is looking for, or when the term the user enters has several meanings. The system that we will develop will help the user explore interactively the repository and the terms that are most significant in this repository.

1.2 The Problem Domain

The project is designed for users who do not know exactly what they are looking for in a repository, so they find it difficult to describe the topic that interests them.The program will allow such people to effectively and conveniently navigate the database, through an interactive process combining search and browse.

The input of the system is:

1. Document database updated regularly (in batches of documents)
2. The text of the documents.
3. Metadata for each document. Metadata are fields of information such as document name, author, date, keywords.

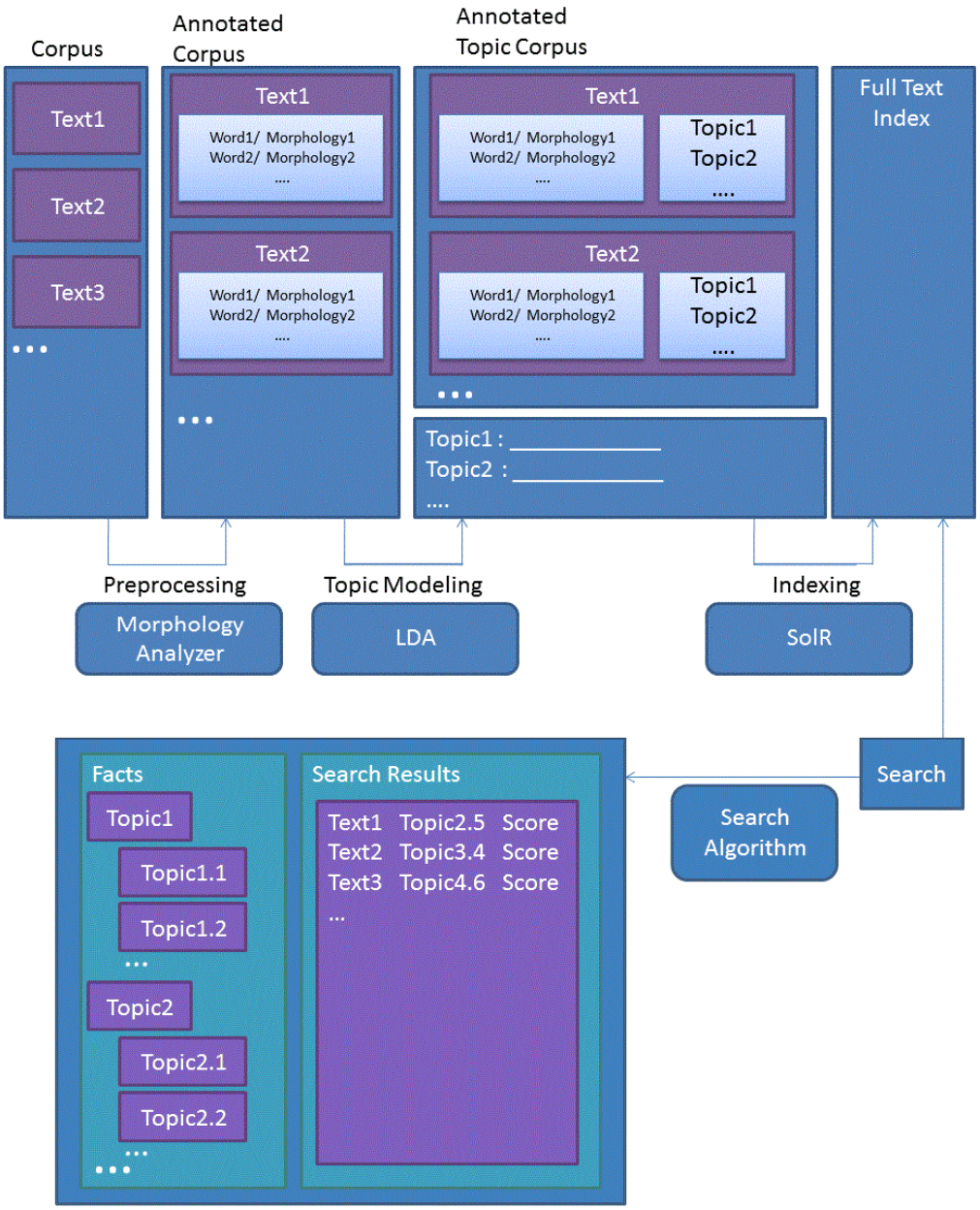
According to these data, build the repository. The repository has several components:

1. Full text index
2. Data base that keeps all the data (documents, metadata and terms)
3. Topic model: a topic model is learned automatically from the documents using a specialized algorithm called LDA. The topic model includes two components:
    - 3.1 A set of topics which capture the most important types of information in the repository.
    - 3.2 A probabilistic model which determines the set of active topics given a document.

Finally, given the repository, our system will include a Graphical User Interface (GUI) which will allow the user to interactively explore the repository using a combination of search and browse operations. Search will use the full-text index. Explore will use the topic model.

The system has as specific objective to support indexing and search of documents written in Hebrew.

The following diagram illustrates the structure of the system.



1.3 Stakeholders

People with relevant products influence include:
1. Researchers interested in the algorithm for learning topic models given a text repository (Dr. Michael Elhadad, Dr. Menachem Adler and Mr. Rafi Cohen)
2. The customer is the Ministry of Science who is funding a research project on this topic
3. We have two types of end users in two domains: (a) the domain of Jewish Law (Halacha) – interested users areHebrew-speakingpeople interested in Jewish law in everyday life. The program can be used in schools for religious education. (b) The domain of medicine: the repository includes questions and answers in the field of public health.

We and researchers will be responsible for the initial design of the system. When we run tests on the beta users, uses them to improve the design if necessary.

1.4 Software Context

The system builds on several existing software modules:

SolR/Lucene: this is a Java-based server which can build a full-text index given a collection of text documents and supports search.

Dr. Adler's Hebrew Morphological Analyzer, which receives text in Hebrew, divides the text into words and classifies each word according to its syntactic role.

LDA –Algorithm: this algorithm learns a topic model given a collection of texts. Our system should allow to change it easily, with suitable permissions, in the future.

Topic Model - using LDA to return relevant topics to the given document.

In addition to these modules, we have available three document corpora:

Corpus of Halachic text in Hebrew that includes question and answers (She'elotv-tshuvot) from various historical periods. The corpus is annotated by experts with rich metadata, and in particular includes an ontology of halachic concepts (that is, documents are tagged by topics manually entered and organized in a hierarchy).

Medical domain: a corpus of texts in Hebrew extracted from the Infomed.co.il health-question answering site. This corpus contains texts that are questions asked by users and answers provided by medical doctors.

Wikipedia in Hebrew: a dump of the Wikipedia site in Hebrew.

Wikipedia in English: a dump of the Wikipedia site in English.

Given these components, we intend to develop:

- Advanced learning algorithms to construct efficient topic models in Hebrew texts that take advantage of existing metadata.
- GUI - user-friendly and convenient graphical interface for users to search and explore material in the repositories

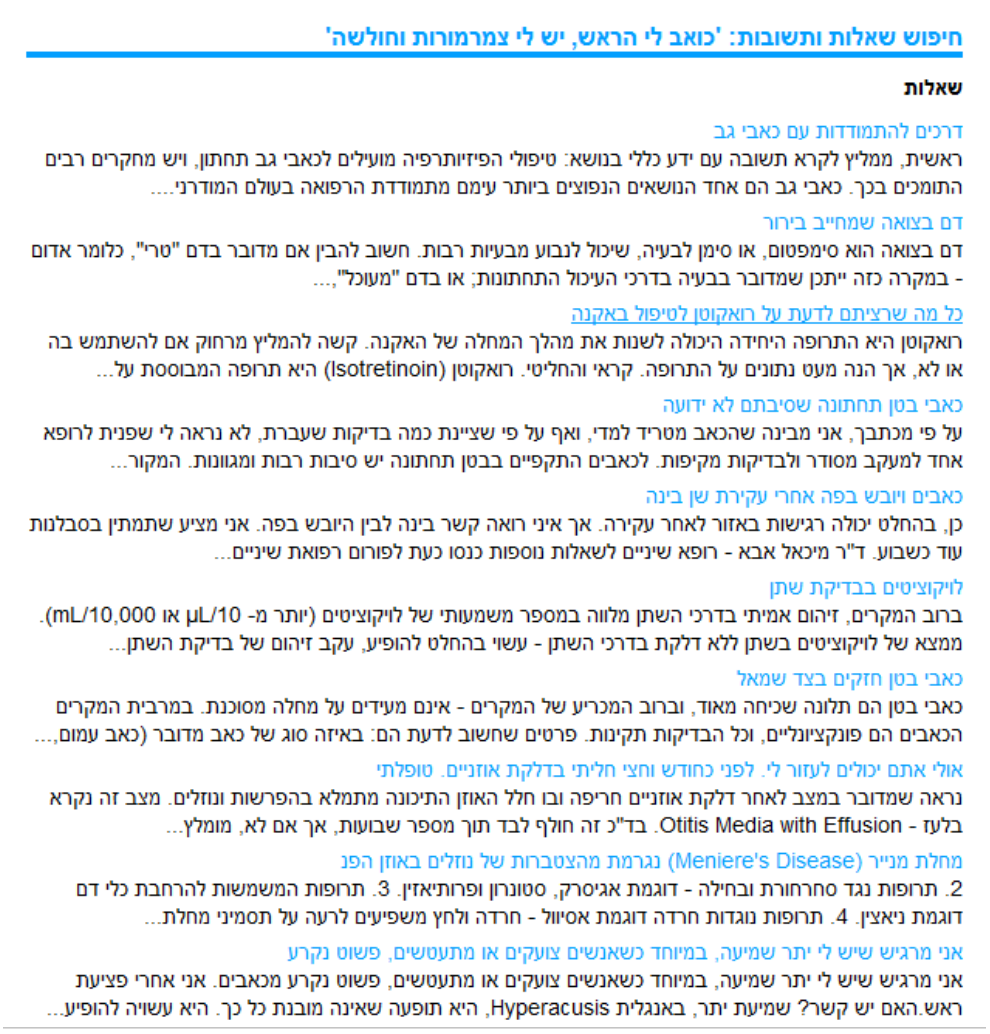
- Every corpus would run on its own computer, with its own URL.

One of the corpuses we have is the medical corpus. This corpus based on the site http://www.infomed.co.il/, which looks like this:

At this site, when a user wants to find something he can:

1. Enter a query for the all portal or just at the forums, search for a doctor, medical institutes, dentistry and many more options.
2. Navigate the site in the forums or in the portal, at the news, doctors, medical institutes, diseases, tests and so on…

Example of results to a given query:



חיפוש שאלות ותשובות: 'כואב לי הראש, יש לי צמרמורות וחולשה'

שאלות

דרכים להתמודדות עם כאבי גב
ראשית, ממליץ לקרא תשובה עם ידע כללי בנושא: טיפולי הפיזיותרפיה מועילים לכאבי גב תחתון, ויש מחקרים רבים התומכים בכך. כאבי גב הם אחד הנושאים הנפוצים ביותר עימם מתמודדת הרפואה בעולם המודרני....

דם בצואה שמחייב בירור
דם בצואה הוא סימפטום, או סימן לבעיה, שיכול לנבוע מבעיות רבות. חשוב להבין אם מדובר בדם "טרי", כלומר אדום - במקרה כזה ייתכן שמדובר בבעיה בדרכי העיכול התחתונות; או בדם "מעוכל",...

כל מה שרציתם לדעת על רואקוטן לטיפול באקנה
רואקוטן היא התרופה היחידה היכולה לשנות את מהלך המחלה של האקנה. קשה להמליץ מרחוק אם להשתמש בה או לא, אך הנה מעט נתונים על התרופה. קראי והחליטי. רואקוטן (Isotretinoin) היא תרופה המבוססת על...

כאבי בטן תחתונה שסיבתם לא ידועה
על פי מכתבך, אני מבינה שהכאב מטריד למדי, ואף על פי שציינת כמה בדיקות שעברת, לא נראה לי שפנית לרופא אחד למעקב מסודר ולבדיקות מקיפות. לכאבים התקפיים בבטן תחתונה יש סיבות רבות ומגוונות. המקור...

כאבים ויובש בפה אחרי עקירת שן בינה
כן, בהחלט יכולה רגישות באזור לאחר עקירה. אך איני רואה קשר בינה לבין היובש בפה. אני מציע שתמתין בסבלנות עוד כשבוע. ד"ר מיכאל אבא - רופא שיניים לשאלות נוספות כנסו כעת לפורום רפואת שיניים...

לויקוציטים בבדיקת שתן
ברוב המקרים, זיהום אמיתי בדרכי השתן מלווה במספר משמעותי של לויקוציטים (יותר מ- 10/µL או 10,000/mL). ממצא של לויקוציטים בשתן ללא דלקת בדרכי השתן - עשוי בהחלט להופיע, עקב זיהום של בדיקת השתן...

כאבי בטן חזקים בצד שמאל
כאבי בטן הם תלונה שכיחה מאוד, וברוב המכריע של המקרים - אינם מעידים על מחלה מסוכנת. במרבית המקרים הכאבים הם פונקציונליים, וכל הבדיקות תקינות. פרטים שחשוב לדעת הם: באיזה סוג של כאב מדובר (כאב עמום,...

אולי אתם יכולים לעזור לי. לפני כחודש וחצי חליתי בדלקת אוזניים. טופלתי
נראה שמדובר במצב לאחר דלקת אוזניים חריפה ובו חלל האוזן התיכונה מתמלא בהפרשות ונוזלים. מצב זה נקרא בלעז - Otitis Media with Effusion. בד"כ זה חולף לבד תוך מספר שבועות, אך אם לא, מומלץ...

מחלת מנייר (Meniere's Disease) נגרמת מהצטברות של נוזלים באוזן הפנ
2. תרופות נגד סחרחורת ובחילה - דוגמת אגיסרק, סטונרון ופרותיאזין. 3. תרופות המשמשות להרחבת כלי דם דוגמת ניאצין. 4. תרופות נוגדות חרדה דוגמת אסיוול - חרדה ולחץ משפיעים לרעה על תסמיני מחלת...

אני מרגיש שיש לי יתר שמיעה, במיוחד כשאנשים צועקים או מתעטשים, פשוט נקרע
אני מרגיש שיש לי יתר שמיעה, במיוחד כשאנשים צועקים או מתעטשים, פשוט נקרע מכאבים. אני אחרי פציעת ראש.האם יש קשר? שמיעת יתר, באנגלית Hyperacusis, היא תופעה שאינה מובנת כל כך. היא עשויה להופיע...

We think that this is very confusing for a user that doesn't know exactly what he is looking for. Even if he enters a query for the all portal, he can get so many references without the exact topics for each reference.
At our system, a user can enter a long text as query, and get a list of results. Each result would have topics and its metadata (in this case the: title, medical field, date, questioner's name, replier's name). We think that this would help the user to understand the results better. In addition, a hierarchy of relevant topics tree would be next to the search results, and the user can navigate at this tree and get a better picture of the repository in the system.

Another corpus we have is Halachic text in Hebrew corpus. This corpus consists of a lot of Halacha's articles. The metadata would be who wrote the article.
We will insert two more corpuses:

1. Wikipedia in Hebrew: a dump of the Wikipedia site in Hebrew.
2. Wikipedia in English: a dump of the Wikipedia site in English.

1.5 System Interfaces

Corpus - Contained from a lot of text files.
Morphology Analyzer - An external library that gets a text and returns this text with his words annotated by their part of speech rule. We will use it in order to annotate all the texts in the corpus.
Annotated Corpus - The same original corpus with annotations by the Morphology Analyzer.
LDA - An algorithm that gets a set of texts, learns the topics from the texts and divides those topics into a hierarchy topics tree. We will use it to add to each text topics and to get the hierarchical topics tree of the all texts in the corpus.
Annotated Topic Corpus - The same original annotated corpus with topics to each text file, and the hierarchy topics tree.
SolR - An external library that gets a query in a specific form and return answer according to the texts in the database. It would help us to find results, according to the corpus, to a user's query.
Full Text Index - The same original Annotated Topic Corpus just in the form of SolR's database. SolR builds thisFull Text Index so SolR can search in it the data SolR needs.
Search - The Graphical User Interface (GUI) for search, here the user can enter his texts as queries.
Search Algorithm - The algorithm we will use to find the results of the users' queries.
Search Results - The Graphical User Interface (GUI) for displaying the results of the users' queries. The form of each result would be:

- a reference to the text file from the original corpus (without annotations or topics),
- the name of the text's topics,
- and the score "how much the text fits the query?"

The user can choose the references that he wants and get the full text files.
Facts - The Graphical User Interface (GUI) for displaying a hierarchy topics sub-tree which includes only the relevant topics for the query. The user can choose the topics that he wants and that way to navigate and understand the corpus in an easy way.

1.5.1 Hardware Interfaces

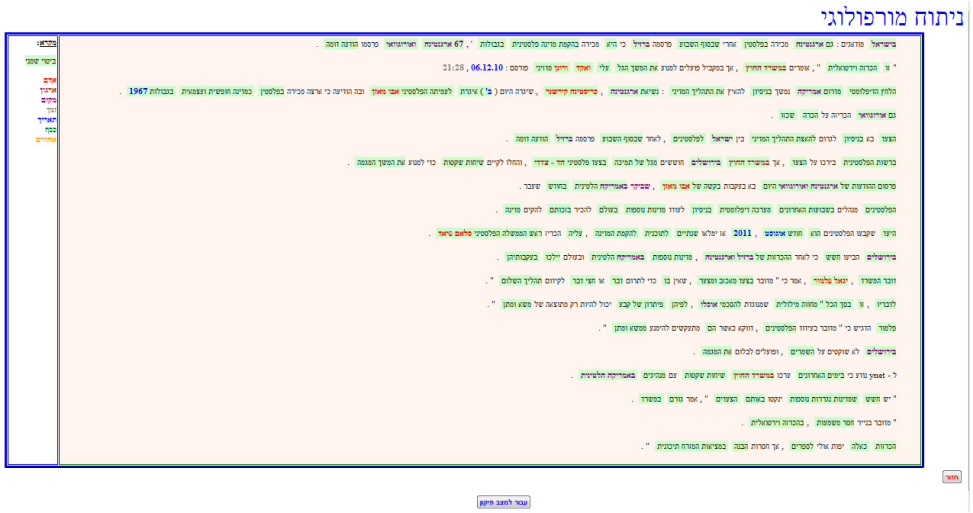
We don't have any hardware interfaces.

1.5.2 Software Interfaces

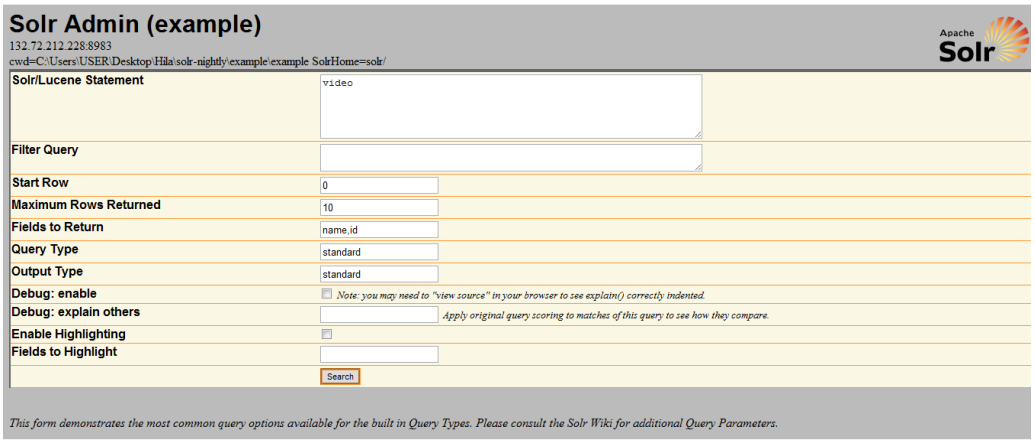
We use the libraries listed above (SolR, Morphological Analyzer, LDA algorithm).
In addition, we will use a SQL database to store the topic models and the documents metadata.

Morphological Analyzer is an external library that gets as input UTF8-text file and returns the same file annotated. The annotation denotes for every word, it's part of speech, if it is singular or plural, and it's gender.
Our system uses this analyzer to annotate all the text in the repository.
The input screen:

The output:



SolR is an external library that gets as input query with specific format and returns text that suite the query. The text that SolR returns is a text that written in the XML files at the database. We can change/delete these files and we can add new ones. SolR knows how to work with annotated XML files, in this case the annotation have been done by the morphology analyzer.
Our system adds all the files in the repository as the SolR's database. When a user submits a text as query, the system analyzes the given text, and enters the needed info from the text as an input query at the right format to SolR. Our system will use the SolR's output in order to answer to the user.
The input screen:



The output:

0 1 name,id on 0 video standard standard 2.2 10 MA147LL/A Apple 60 GB iPod with Video Playback Black EN7800GTX/2DHTV/256M ASUS Extreme N7800GTX/2DHTV (256 MB) 100-435805 ATI Radeon X1900 XTX 512 MB PCIE Video Card

LDA algorithm is an algorithm that gets a set of texts, learns the topics from the given set of texts, and divides those topics into hierarchal topics tree. In the future, this algorithm might be change, therefore we need to support in changing it.

Topic Model is a process that uses the LDA algorithm to understand who the important topics are. It gets texts and returns the topics.

1.5.3 Events

The key events are:
End-user writes a text as query.

End-user navigates the search results.

The administrator creates a new document repository.
The administrator updates an existing document repository.
The administrator deletes an existing document from the repository.

The system records how it is used by end-users, and learns how to update the internal topic model based on this feedback. In addition, user statistics are collected, and the system tries to produce user-specific replies.

2 Functional Requirements

Database Preprocessing: the system has corpus with text files. After the morphology analyzing phase these files become annotated files. After the topic modeling phase the hierarchy topics tree is built based on the topic modeling, and the suitable topics are added for each file. The indexing procedure is needed for inverting the corpus to a more convenient form for searching.

End users: the user enters a query in natural language, the system analyzes his query, uses SolR and search algorithms (using the metadata and the topics) to find answer to the user.The user gets an answer that helps him navigate and understand the structure of the corpus. The answer would be a list of references to relevant texts from the repository, for each reference we would display its topics and its metadata. In addition, the answer includes a hierarchy with the relevant topics tree – this tree would help the user navigate and understand the structure of the corpus, and it shows only a part of the answer of a query.

Content manager: gets new documents and add them to a repository.

Administrator: gets statistics from the use of the system, learn from them and update the hierarchical division to topics accordingly.
The statistics:

- How many times a topic has been selected in searches.
- How many times a document has been selectedin searches.
- How many times a document and a topic has been selectedtogether in searches.

After building the repository, we have several text files. Each text file has its original text, its relevant topics, annotation and the metadata. In addition, we have a hierarchy topics tree that would built according to the topic model using an algorithm called LDA. In this tree, every node is a topic from the topic model, the leafs are the texts that relevant to their parent node's topic. Every node's topic is a subtopic of the node's ancestor's topic.
Building the repository is a preprocessing phase; it's done before the user enters a query.

3 Non-functional requirements

3.1 Performance constraints

Speed, Capacity & Throughput:

- Every transaction must be processed in no more than 2 seconds.
- 10,000 users can use the system simultaneously.

Reliability.

At 95% cases the system can find the relevant data the user is looking for.

Portability.

The system can run on every operation system.

The system is Web service, but it doesn't depend on specific Web browser.

The system should support the text in English and in Hebrew.

Usability.

The user should understand how to use the system in 10 minutes.

The user can prepare the input in 10 seconds.

The user can interpret the output in 2.5 minutes.

Availability.

The system should be available 100% of the time, depends on the availability of the hardware and the network.

3.2 Platform constraints

All the existing software modules are written in Java, which will make it much easier to interact with them if we program in Java as well.

3.3 SE Project constraints

The system works interactively with the users.
The inputs come from the users.
At our simulation, we will use beta-users that will use the system and will help us make it more user-friendly.
We will implement the system in Java, and check it by beta-users.
We will implement the system on our and the university computers.

The system will look something like this:

An example for results to a given query:

תוצאות עבור החיפוש "כואב לי הראש, יש לי צמרמורות וחולשה":

כאב ראש חזק ומתמשך שמצריך בירור.
כל מה שרציתם לדעת על: אני בת 29, בריאה בדרך כלל, למעט ניתוחים אורתופדיים שעברתי ביד בעקבות שבר. בחודש האחרון התחילו להופיע אצלי כאבי ראש בעוצמה מאוד חזקה.
נושא: כאבי ראש כרוניים, ניתוחים אורתופדיים בגפיים

כאבי ראש בילדים
התלונה על כאבי ראש שכיחה אצל ילדים והיא עלולה להופיע כבר בגיל שנתיים, ללא קשר עם הופעת חום או מחלה נלווית אחרת. כאב ראש מתמשך אצל ילדים...
נושא: כאבים שונים אצל ילדים, תופעות לוואי של תרופות לילדים, כאבי ראש כרוניים

עץ נושאים:

- רפואת ילדים
  - כאבים שונים אצל ילדים
  - מחלות נגיפיות בקרב ילדים
  - חום אצל ילדים
- כאבים כרוניים
  - כאבי ראש כרוניים
  - כאבי בטן חוזרים

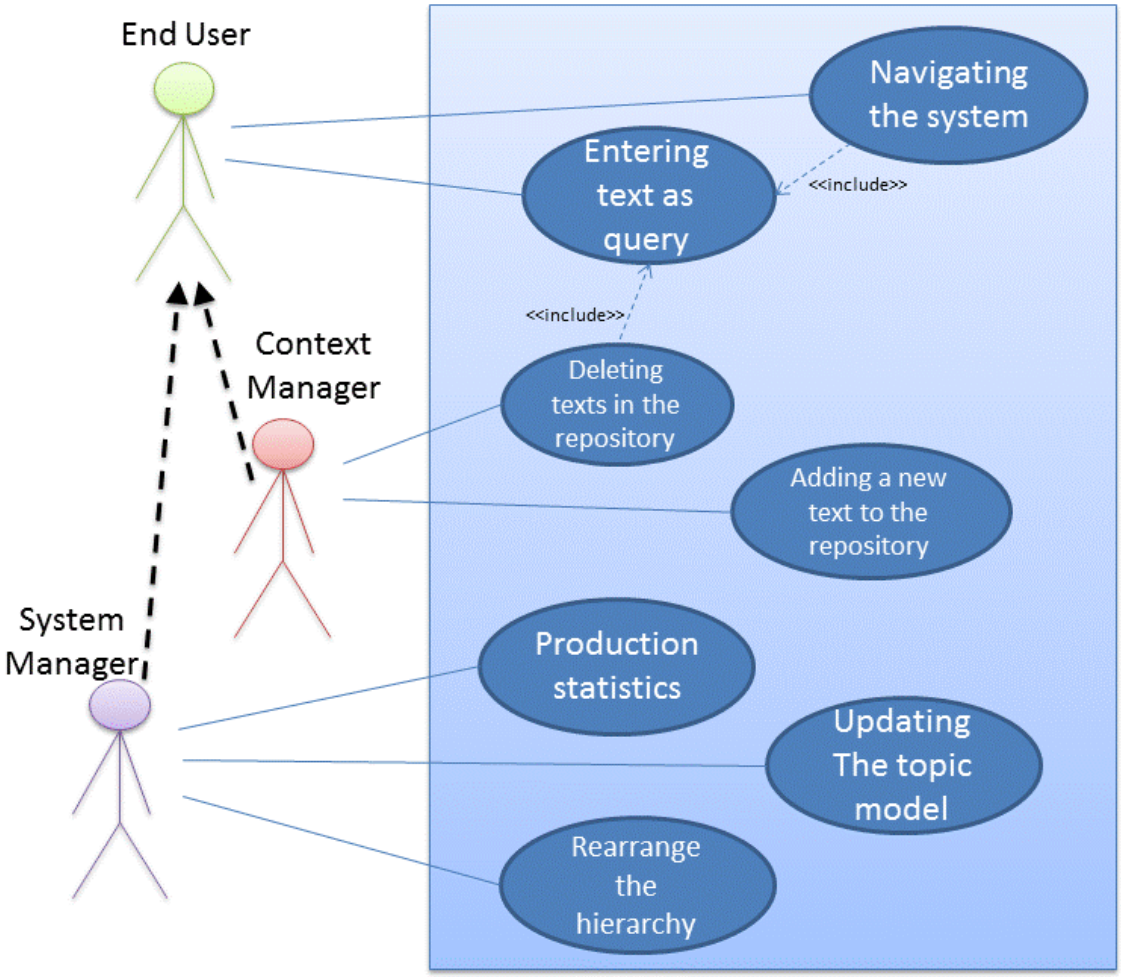
4 Usage Scenarios

1. Entering a text: the user enters a text or words that are related to the subject he is looking for. The system tries to associate this text or these words to one of the what-her existing topics. According to the system topic found, returns an answer to the user.
2. Navigation system: the user goes on the list of topics, choose topic that seemed relevant. If this topic to subtopics process can continue. So until the user comes to the topic he is looking for.
3. Collecting statistics: the system administrator generates a report statistics on the use of the system.
4. Updating the topic models: the system administrator updates the topic models.
5. Updating the repository: Content Manager updates the knowledge base.

4.1 User Profiles – The Actors

1. There are three types of players:
   End users - enter text/queries and navigate the system. Than utilization for purposes of these system collects statistics.
2. Content Manager - modifies the system's database.
3. Administrator - uses statistics gathered by the system.

4.2 Use-cases


End User
Context Manager
System Manager
Navigating the system
Entering text as query
<<include>>
<<include>>
Deleting texts in the repository
Adding a new text to the repository
Production statistics
Updating The topic model
Rearrange the hierarchy

Use Case 1
Name: Entering a text as query.
Primary Actor: The end-user
Pre-Condition: The application is running. The preprocessing has been done successfully. The repository is not empty.
Post-Condition: The system returns an answer to the user. The answer is like what described above.
Description: The user can enter a text as query for the purpose of searching some data he is looking for.
Flow of events:

1. The user entered the application.
2. The user enters a text as a query
3. The user submits the query.
4. The system processing the query.Tr
5. The system searching for the query's relevant data.C
6. The system presents to the user an answer, as described above.
7. The system saves the user's use (the texts and topics) at the statistics.
8. The user can choose one or more of the texts that are part of the answer.

Alternative flows: Instead of choosing texts, the user can navigate the system by choosing topics that are also part of the answer, or enter a new query (back to 3).

Use Case 2
Name: Navigating the system.
Primary Actor: The end-user
Pre-Condition: The application is running. The preprocessing has been done successfully. The repository is not empty. The user submitted a query and got an answer (by the use case "Entering a text as query").
Post-Condition: The system shows to the user the hierarchy topics tree. The hierarchy topics tree is like what described above.
Description: The user can navigatethe system by navigating the hierarchy topics tree for the purpose of searching some data he is looking for.
Flow of events:

1. The user chooses one of the topics.
2. If the topic is not a leaf in the tree, the system shows to the user the topic's subtopics (back to 1).
3. If the topic is a leaf in the tree, i.e. the texts of this topic, the system shows to the user the full references texts list.
   3.1. The user can choose the references to the texts that he wants.

Alternative flows: at any point the user can choose one or more from the given texts, or enter a new query.

Use Case 3
Name: Producing statistics
Primary Actor: System manager
Pre-Condition: The application is running. The user is registered to the system as the system's manager.
Post-Condition: The system returns the relevant statistics to the user.
Description: The user can produce statistics about the system.

These statistics:

- How many times a topic has been selectedin searches.
- How many rimes a document has been selectedin searches.
- How many times a document and a topic has been selectedtogether in searches.

Flow of events:

1. The user choosestoproduce statistics.
2. The system calculates and returns to the user the described above statistics.

Use Case 4
Name: Updating the topic model.
Primary Actor: System manager
Pre-Condition: The application is running. The user is registered to the system as the system's manager.
Post-Condition: The topic model has been updated.
Description: The system manager can update the topic model by changing the algorithm of topic modeling.
Flow of events:

1. The user chooses to update the topic model.
2. The system shows to the user the current topic model.
3. The user changes the topic model.
4. The user submits the changes.

Use Case 5
Name: Rearrange the hierarchy.
Primary Actor: System manager.
Pre-Condition: The application is running. The user is registered to the system as the system's manager.
Post-Condition: The hierarchy topics tree has been rearranged.
Description: The system manager can update the hierarchy topics tree. He can add/remove/change topics names/change the hierarchy/… of the hierarchy topics tree.
Flow of events:

1. The user chooses torearrange the hierarchy topics tree.
2. The system shows to the user the current the hierarchy topics tree.A
3. The user changes the hierarchy topics tree.
4. The user submits the changes

Use Case 6
Name: Adding a new text to the repository.
Primary Actor: Content manager
Pre-Condition: The application is running. The user is registered to the system as the content's manager.
Post-Condition: The new text has been added to the repository, annotated and with the suitable topics. The hierarchy topics model tree is updated respectively.
Description: The user can add new text files to the repository.
Flow of events:

1. The user chooses to add a new text file.
2. The user submits a text file.
3. The system adds this text to the corpus, and does the database preprocessing again (the morphology analyzing, the topic modeling and the indexing).

Use Case 7
Name: Deleting a text from the repository.
Primary Actor: Content manager
Pre-Condition: The application is running. The user is registered to the system as the content's manager. The user found the text he wants to delete (by the use case "Entering a text as query").
Post-Condition: Thetext has been deleted.The hierarchy topics model tree is updated respectively.
Description: The user can delete existing text from the repository.
Flow of events:

1. The user chooses to delete the selected text.
2. The system removes this text from the corpus, and does the database preprocessing again (the morphology analyzing, the topic modeling and the indexing).

5 Appendices

.

The search algorithm that we will implement is from this book:
http://nlp.stanford.edu/IR-book/information-retrieval-book.html
The topic modeling that we will implement is from this paper:
http://psiexp.ss.uci.edu/research/papers/SteyversGriffithsLSABookFormatted.pdf
The score that we will calculate would be by methods that this book offers:
http://nlp.stanford.edu/IR-book/information-retrieval-book.html