Grant Agreement Number: 257528

KHRESMOI

www.khresmoi.eu

Report on summarization techniques

Deliverable number	*D4.4*
Dissemination level	*Public*
Delivery date	*28 February 2013*
Status	*Final*
Author(s)	*Liadh Kelly, Johannes Leveling, Shane McQuillan, Sascha Kriewel, Lorraine Goeuriot, Gareth Jones.*

This project is supported by the European Commission under the Information and Communication Technologies (ICT) Theme of the 7th Framework Programme for Research and Technological Development.

Abstract

In this document we describe a summarizer for text documents, developed at Dublin City University (DCU), Ireland for the Khresmoi project. The aim of the summarizer is to provide a summarized view of medical documents for use in the Khresmoi system interface. To achieve this the summarizer selects the most meaningful/interesting segments in a text, for inclusion in the summary, by using features to describe segments and weight the importance of segments in documents. In this document we describe the summarizer, its functionality, use in Khresmoi, and evaluations conducted to investigate its utility in summarizing medical documents and in improving the user experience with the Khresmoi prototype.

Table of Contents

List of Abbreviations

CNGL	Centre for Next Generation Localisation
DCU	Dublin City University
ezDL	Easy Access to Digital Libraries
HON	Health on the Net
MIMIR	Multi-paradigm Indexing and Retrieval
TF-IDF	Term Frequency Inverse Document Frequency
TS-ISF	Inverse Sentence Frequency
UMLS	Unified Medical Language System
URL	Universal Resource Locator

1 Executive Summary

Searching for information is often part of a larger activity of busy individuals for whom the volume of material to be inspected may be overwhelming, or on other occasions individuals may simply be too lazy to inspect the information returned by a search engine. In order to move ahead with their wider activities, when seeking information from medical collections users want to arrive at the required information as quickly and effortlessly as possible. The overall aim of the Khresmoi system is to meet these user needs. As part of this, the Khresmoi system includes a summary of items of potential interest to a user which seeks to improve the effeiency of information access. In this document we describe the summarizer used to generate these summaries.

A good document summary should contain the most important information from a document in a form coherent to the user of the information. Achieving this is challenging for many reasons including the very varied forms of the information retrieved, and the uncertainty of knowing exactly what the user's information needs are. Taking these challenges, the Khremoi summarizer attempts to achieve this by selecting the most meaningful/interesting segments in a text using features to describe and weight the importance of segments in documents. The highest weighted segments (sentences in our case) are used to form the document summary.

Initial experiments were conducted on the summarizer described in this document to evaluate its utility in improving the user experience in the Khresmoi system. Results of these experiments show potential for the summarizer, but also indicate that further refinement is required to support a user in locating required information to meet their needs as expressed through their queries. Ongoing development is exploring new feature development for the summarizer, and specifically adapting the summarizer to medical data used in Khresmoi (medical articles and abstracts).

2 Introduction

The Khresmoi prototype supports retrieval of required medical information in response to a user query. To express their information need to the Khresmoi prototype, the user of the prototype interface (the Khresmoi ezDL Swing prototype interface is shown in Figure 1) is required to enter query terms, and optionally use provided filtering functionality (e.g., filter based on the classification assigned to documents). A list of results is presented to the user in response to their query (as shown in the region marked (1) in Figure 1). When the user chooses a result from the result list, further details on this item are shown in the details view (region marked (2) in Figure 1). The details view aims to provide precise succulent details to the user on the selected item, thereby allowing the user to quickly gain an overview and understanding of the document's full contents. The idea is that in several cases, this will provide the user with sufficient information to either know that the item is not relevant to their information need or, wholly or partically, answers their information need; and in the remaining cases it will provide them with enough information to know that the result item will be interesting for them to view (this can be done by clicking on the provided link to the full document content in the details view). As part of the process of providing concise succulent information relating to an item in the detailed view, a summary of the item's content is provided. This summary is generated using the DCU summarizer. Figure 2 shows a sample of the type of summary generated by the summarizer. In the figure we can see an original piece of text from the Khresmoi medical index and the summary generated by the summarizer for this text.

The DCU Summarizer is an automatic summarization tool built at the Center for Next Generation Localisation (CNGL), DCU. It uses sentence extraction to generate summaries. The idea is that the most important sentences in a document (i.e. the ones in the final summary) can be used to represent the original text. The selection methods are based on Information Retrieval and Natural Language Processing techniques. The current version of the summarizer is tuned to summarize documents in the

English language, as described in Section 3. Other languages will be supported in future iterations of the summarizer.

Section 3 and 4 provide further details on the summarization approach adopted. Specifically, Section 3 provides details of the summarizer architecture, and the configurable components of the summarizer are described in Section 4. Evaluations to determine the utility of the summarizer are presented in Section 5.

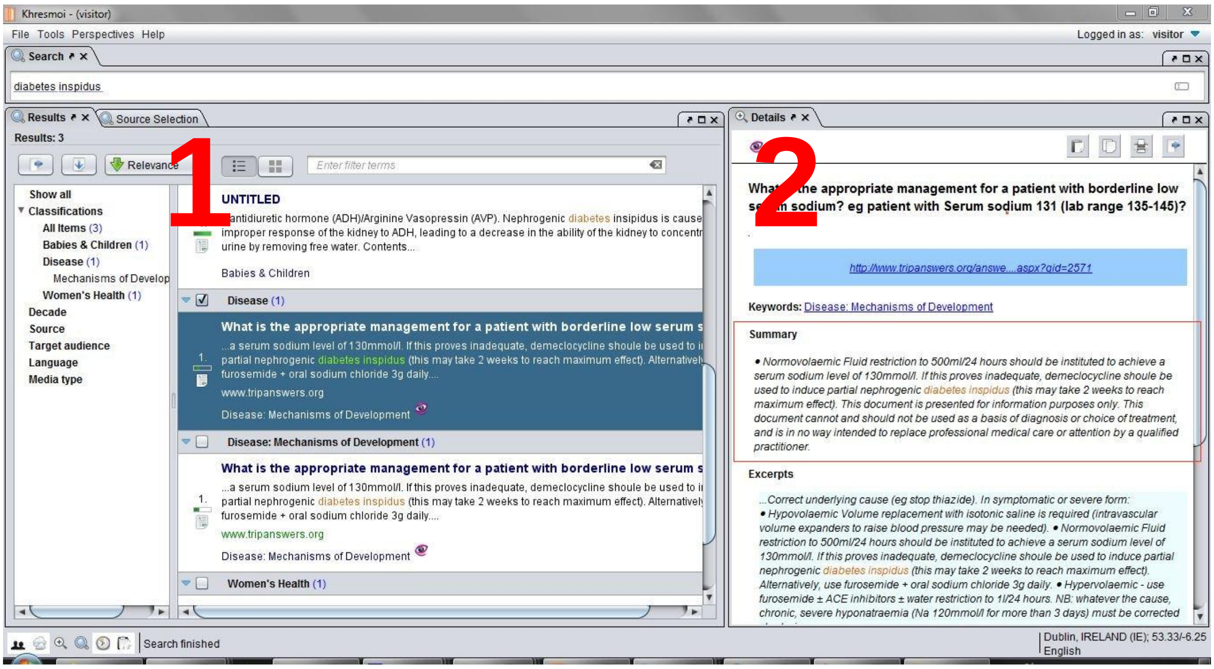

Figure 1: ezDL prototype with list of results (1) and summarizer integrated, as highlighted, in the detail view (2).

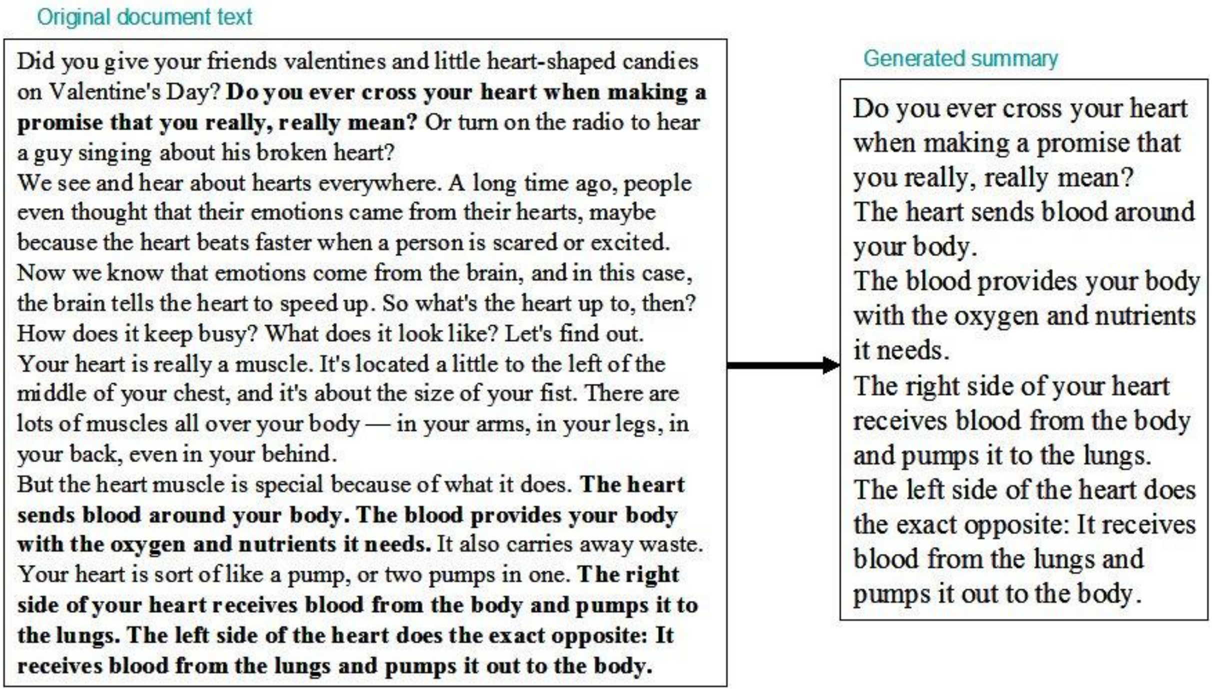

Figure 2: Summarizer in operation.

3 Summarizer Architecture

The summarizer is written in Java and uses the following external libaries: Apache Lucene 3.6.1, Apache Commons Configuration 1.8, Apache Commons IO 2.4, Apache Commons Lang 2.6, Apache Commons Logging 1.1.1. It takes as input a URL or raw text, and produces as output a summary of the text. The summary generation is a four step process (shown in Figure 3): 1) document segmentation, 2) feature extraction, 3) segmentation classification, and 4) abstract generation.

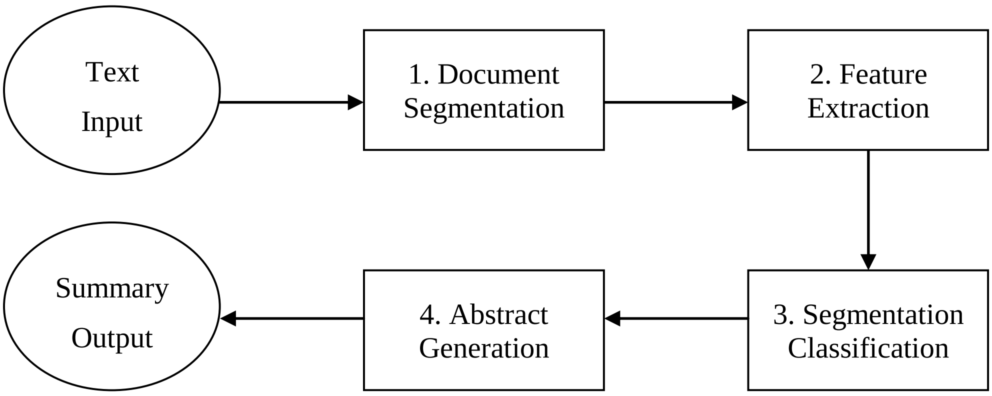

Figure 3: Summarizer architecture.

The first step in the summarization process is to separate the document into different segments. To do this the input text is split into different linguistic units, such as paragraphs, sentences and words. Thus, the tokenization includes tasks such as paragraph detection (e.g. based on text markup such as <p>), sentence boundary detection, and phrase recognition (e.g. based on frequently occurring word n-grams).

The second step in the summary generation process is feature extraction, where features are extracted from the text and scores assigned to them. For example, scores assigned to words (e.g. for capitalization), phrases (e.g. occurrence in a specific word list or lexicon), or the sentence (e.g. sentence length). To conduct this process sentences are indexed using Lucene, where each sentence is indexed as a unique Lucene document. Features are described in detail in Section 3.1.

Thirdly, for each sentence in the main body of a text, its score is computed as a weighted sum of feature scores. That is, for each sentence, an aggregated score is computed which is based on the scores assigned to individual word, phrase and sentence features. The resulting score is then normalized into [0...1]. Finally, in step four, the highest scoring sentences are used to generate the document summary.

In the first iteration of the summarizer, described in this document, we used a set of sixteen features to indicate the importance of text segments for the summary. These features are described next, in Section 3.1. Weights for these features were manually chosen based on empirical observation[1].

3.1 Feature Description

We investigate different sentence features that have proved useful in various automatic text summarization systems. To easily describe features used in the summarizer, we divide all features into two groups: Term Checking Features and Non Term Checking features (i.e. all other features).

3.1.1 Term Checking Features

Term checking features are based on a check of whether a text fragment (e.g. a sentence or paragraph) contains words or phrases with certain characteristics. The characteristics can relate to word occurrence in a list, capitalization information, etc. For example, the title of an article often reveals the major subject of that document. Sentences containing terms from the title are likely to be good summarization candidates. A typical feature belonging to this group is the Title Term Feature. We used many features that have been proposed in previous research [1-3] and are described in some surveys of Text Summarization Extractive Techniques [4]:

- **Basic Words features:** This feature models word complexity and may be useful to produce summaries for non-expert users that are not overly familiar with more technical language [4].
- **Cue Phrase Feature:** This feature checks for specified indicator phrases such as "importantly" or "in summary" in a sentence. These phrases may indicate that the sentence is good for summarization (positive phrases), or that it is not (negative phrases). This is one of the most important features in extractive summarization systems [4].
- **Date/Time Feature:** This feature checks for temporal expressions (e.g. weekdays or month names) in a sentence. It was used in the evaluation in [5].
- **Named Entity Feature:** Calculates the number of named entities that occur in each sentence. As named entity taggers are quite slow, a more naive approach is taken here. Any word (except the first in a sentence), that starts with a capital letter is assumed to be a named entity [5]. For English, proper nouns are capitalized and common nouns are not.

[1] Current summarizer development work in Khresmoi aims at employing machine learning to automatically assign weights. This work is currently under review and will be described in a forthcoming publication.

- **Preposition Feature:** This feature tests if prepositions occur in a sentence [5] (using a predefined list of prepositions from the part-of-speech tagged Brown corpus).
- **Pronoun Feature:** Similar to the Preposition Features, this feature uses a predefined pronoun list. Pronouns should not be included in a summary unless the entities they referred to are also included or they are expanded into corresponding nouns [5].
- **Punctuation Feature:** Calculates the proportion of punctuation tokens in all tokens in the sentence. If the value exceeds a specified threshold, the sentence is presumed to contain noise and will be unimportant for a summary.
- **Query term overlap:** High overlap with the query implies importance; e.g. the sentence "Cardiovascular *disease* is a class of *diseases* that involve the *heart* or blood vessels (arteries, capillaries and veins)." includes all terms for the query "heart disease".
- **Location feature:** Sentences under headings or in a summary are more important.
- **Affix feature:** This feature tests if a word contains any one from a list of medical affixes (i.e. word prefixes, in-fixes, and suffixes) in a manually created list of medical affixes for words derived from Latin and Greek. The idea is that medical technical terms are often derived from Latin or Greek and these terms would indicate the importance of a sentence.

The first six features are language dependent, e.g. different noun lists would be used for different languages. The remaining features are language independent, but would also need reweighting for different languages. The current version of the summarizer generates summaries for English text. Extending to other languages should however be straight forward (i.e. by changing the configuration, see Section 4).

3.1.2 Non Term Checking Features

The Non Term Checking Feature group includes features with different methods compared to the above group. These features also have been used in many summarization systems before. These features are also language independent, but would need reweighting for different languages. For a more detailed description of these features, the reader is again referred to the original literature.

- **Cluster Keyword Feature:** Feature proposed by Luhn [6]. Luhn specified that two significant words can be considered as related if they are separated by not more than five insignificant words. Important sentences will have large clusters of significant words.
- **The Global Bushy Feature:** This feature generates inter-document links based on similarity of paragraphs; paragraphs with many links share vocabulary with many other paragraphs and are important; proposed by Salton [7].
- **Number of Terms Feature:** The number of terms in a sentence, assuming that too long or too short sentences are unimportant for a summary [8].
- **Sentence Length Feature:** The count of the number of terms in a sentence, using the term number cut-off instead of string length cut-off.
- **Skimming Feature:** The position of a sentence in a paragraph. The underlying assumption is that sentences occurring early in a paragraph are more important for a summary [4, 9].
- **TS-ISF Feature:** Similar to TF-IDF, but works on the sentence level. Every sentence is treated like a document. Sentences which have a lot of keywords are likely included in summary [4, 10].

4 Configurable Components

Numerous components of the summarizer are configurable. In this section we provide a detailed explanation of each option.

4.1 Information Files

Several features used to score sentences require files containing lists of words or phrases. Different lists may work better for different domains, so these are configurable.

4.1.1 Stopwords

Used during the sentence indexing and tokenization phase. Any words listed in the stopwords file will be ignored.

4.1.2 Cuephrases

The Cue Phrase Feature scores sentences containing phrases listed in the cue phrase file. These can be either positive or negative scores.

`Format: <phrases>,<score> e.g. to conclude,5`

4.1.3 Affixes

The affix presence feature scores sentences containing affixes from the affixes file positively. These can be domain specific.

`Format: <optional hyphen><affix><optional hypen> e.g. -iate`

The hyphen denotes the position of the word's stem.

4.1.4 LocationFeatures

Certain sections contain sentences better fit for use in a summary. The names of these sections are stored along with their scores in the location features file.

`Format: <section>,<score> e.g. introduction,5`

4.2 Multipliers

Some features may prove better for sentence selection in certain domains. It is possible to give specific weights to features in the configuration file, which will either improve that feature's contribution, degrade it, or even cancel it completely (a multiplier of 0). Any value can be used for the weight, i.e. a real value between 0 and 1, or a binary multiplier (e.g. 0 or 1).

4.3 Feature Settings

Some features allow the developer to change more specific settings. This gives the developer more control over the inner workings of those features.

4.3.1 AffixPresence

extraLettersForMatch

To count as a definite match, a predefined minimum number of characters must exist at the appropriate end of a discovered affix.

```
Example:

Affix: -iate

Word: asphixiate

Number of letters: 6.
```

If the number of letters required is 6 or less we have a match.

Default: 3

4.3.2 ClusterKeyword

minSignificantWordSeperation

To be considered a cluster of keywords, significant words must be no greater than a predefined number of insignificant words apart.

```
Example:

Significant words: scoring, information, structural

Sentence: The sentence [[scoring process utilises information] both from the structural] organization.
```

If the minimum separation is 3 we have a cluster of 3 significant terms. If it is 2, the largest cluster has 2 significant terms.

Default: 5

4.3.3 GlobalBushy

minimumSimilarity

Any sentence comparisons below this level will not be counted. They are deemed to be too dissimilar. These low scores are not good indicators of central nodes.

Default: 0.8

maximumSimilarity

Any sentence comparisons above this level will not be counted. Some sentences are almost carbon copies, and will score artificially high. These high scores are not good indicators of central nodes.

Default: 1.0

minimumSentenceQueryLength

To calculate the similarity of each sentence a query is provided to a Lucene index containing all sentences. Sentences with few query terms score artifically high, and are not good indicators of central nodes. Increasing this value improves central node detection, as well as performance (due to a reduction in the number of queries).

Default: 20

4.3.4 Lucene

topTermCutOff

This value specifies which percentage of the term list for the query to discard. The text is loaded into an in-memory index, a sentence per Lucene Document. Then the index is queried for terms and their associated frequency in the index (i.e. for calculating TS-ISF). The *topTermCutoff* is a ratio between 0 and 1 which specifies how far to go down the frequency ordered list of terms ; the terms left for consideration have a frequency greater than *topTermCutoff*topFrequency*. Must be between 0 and 1. The default is to consider all terms if this variable is not set, i.e. a cutoff of 0. But it is recommended to set this to a more appropriate value (such as 0.3).

Default: 0.3

4.3.5 Punctuation

maxRatio

The max ratio of punctuation to terms. If the ratio is higher than this the sentence is marked negatively.

Default: 0.3

4.3.6 ShortSentence

minLength

If the number of terms in a sentence is below this value it is weighted negatively.

4.3.7 LongSentence

maxLength

If the number of terms in a sentence is above this value it is weighted negatively.

4.4 Tokenization

Configuration files are needed at word level and sentence level for tokenization. These are configurable to allow for different languages.

4.4.1 Word Abbreviations

Certain abbreviations can be missed by general rule tokenizers. A list of these can be specified in order to ensure they are discovered.

`Format: <abbreviation>`

4.4.2 Sentence

badStart

A list of bad sentence starts can be listed in the *badStart* file. A sentence cannot start with any of these.

`Format: <bad sentence start> e.g. "!", "...", ")"`

possibleEnd

A list of possible sentence ends can be listed in the *possibleEnd* file. A sentence can end with any of these.

```
Format: <possible sentence end> e.g. "!", "?", "."
```

badEnd

A list of bad sentence ends can be listed in the *badEnd* file. A sentence cannot end with any of these.

```
Format: <bad sentence end> e.g. "("
```

5 Experimentation

A small scale experiment was conducted to examine the utility of the summarizer in the Khresmoi system. In this investigation we looked at the utility of the summaries produced in the ezDL user interface, relative to two other summary generation approaches. These summary generation approaches were: 1) take first three sentences from document as the document summary; and 2) use the snippets returned by MIMIR[2] to generate the document summary (further details on this approach are provided in Section 5.1.1). The experiment setup is described in Section 5.1.1. Experiment results are provided in Section 5.1.2.

5.1.1 Experimental Setup

The purpose of this evaluation is to determine whether summaries can be used to indicate the relevance of documents to user queries. This means that for this experiment we require: 1) a query set, 2) a set of documents which are relevant and a set which are non-relevant to the queries, and 3) summaries for these documents.

5.1.1.1 Query Set

For this small experiment we used a set of 5 general public queries. These 5 queries were randomly selected from a larger set of 50 'long' general public queries[3] created for information retrieval evaluations in the Khresmoi project. The set of 50 'long' general public queries were manually selected by a domain expert (Natalia Pletneva, HON) from a sample of raw queries collected over a six month period from the Health on the Net (HON) search engine[4]. To remove possible influence by web crawlers using predetermined queries, only non-capitalized queries were taken into account. Queries in languages other than English and those which seemed to be too "medical" (for example, complex medical terms) were manually excluded [11]. Only queries with three or more query terms were included in this 'long' queries set[5].

The five queries were:

---

[2] MIMIR is the indexing and retrieval system used in the Khresmoi project and is described at http://gate.ac.uk/mimir/.

[3] In referring to 'long' general public queries, we mean queries generated by members of the general public which contain three or more terms (as distinct from general public queries which contain one or two query terms only).

[4] http://www.hon.ch/

[5] In a similar manner a set of 50 general public queries consisting of queries with one or two terms was also created for information retrieval evaluations, and is described in [11]. In the summarizer evaluation described in this document we opted to use the longer queries which largely had more focused information needs.

1. breast reduction and lift
2. do allergies cause migraines
3. food allergy test
4. head and neck cancer
5. sleep apnea syndrome

5.1.1.2 Document Set

For each of the 5 queries, a set of 5 relevant and 5 non-relevant documents was manually generated with the Khresmoi prototype system using the approach described in this section.

The BM25 retrieval algorithm was used to retrieve a ranked list of documents, from the Khresmoi medical document set, in response to user queries. The top 5 relevant and top 5 non-relevant documents from this result set were manually extracted using the criteria for judging relevance generated for the 'long' general public query set, described in the previous section, by a domain expert (Natalia Pletneva of HON).

5.1.1.3 Summary Generation

For this experiment we compared the summaries generated using the approach described in Section 3, to (1) a simple approach which uses the first three sentences of a document as the document summary (the first three sentences introduce a document's topic and are presumed to form a coherent abstract), and (2) an approach which uses snippets generated by MIMIR to form a document summary. A MIMIR snippet consists of the 75 terms before and the 75 terms after the occurrence of a query term in the document. To generate the summary the first snippet returned by MIMIR for a document is taken. This first snippet is generated from the first sequential occurrence of a query term in a document.

Summaries were generated for the sets of 10 documents, described in the previous section, retrieved for each of the 5 user queries, using the three summary generation approaches. This resulted in sets of 30 summaries being generated for each of the 5 queries.

5.1.1.4 Experimental Procedure

Each of the generated summaries was rated by three computer science researchers in Dublin City University (1 native English speaker). For each of the 5 queries, the generated summaries were presented to each rater in a different random order. For each summary subjects provided answers to the following two questions, on a 5-point Likert scale:

1. How readable is the summary?
2. How relevant is the summary with respect to the query?

In the next section we analyse the results obtained for this evaluation.

5.1.2 Experimental Results

In this section we analyse the results obtained from the experiment. These results are shown in Tables 1, 2 and 3.

5.1.2.1 Question 1: Readability

Table 1 shows the overall average rating across the three subjects for the three different summarization techniques, for the question 'How readable is the summary?'. As can be seen, the Khresmoi summarization technique compared favourably with the excerpts extraction approach (average of 3.84

versus 2.51). However, the technique which used the first three sentences of documents as the summary received a marginally higher readability rating (average of 3.99 versus 3.84). This is perhaps unsurprising given that summaries produced by summarization tools should generally be 10-30% the length of a document, whereas we produced summaries which were a far lower percentage (just three sentences long). Further readability scores most likely were impacted by the presence of erroneous characters in the summaries. These characters were present due to original documents not being pre-processed properly, e.g. some malformed HTML not being removed from web pages.

Tables 2 and 3 show the break down of ratings across documents which were deemed relevant to queries and those which were deemed non-relevant respectively. The same readability ratios are observed here, as those which were observed across the entire collection in Table 1.

5.1.2.2 Question 2: Ability to Determine Document Relevance from the Summary

The second question posed to subjects completing the evaluation looked at the ability of users to determine the relevance of documents to queries from the summaries. Looking at the summaries generated from documents which were relevant to queries, we see that the summaries generated using the Khresmoi summarization approach were most useful in helping subjects determine the documents which were relevant to the queries. This result is shown in Table 2, where for the question 'How relevant is the summary with respect to the query?', the Khresmoi summarization approach obtained an average score across the three subjects of 3.83 (maximum possible score is 5), the first sentences approach received an average score of 3.79, and the excerpts approach received an average score of 3.41.

Despite the positive results obtained for documents which were relevant to the queries, the Khresmoi summarization approach did not prove as useful in helping determine when documents were not relevant to queries. These results are shown in Table 3, where a lower rating to the question 'How relevant is the summary with respect to the query?' is preferable, given that the documents are in fact not relevant to the queries. Here the summary generation approach which takes the first sentences of a document to generate the summary proved most useful. This approach received an average rating across the three subject of 1.60, relative to a rating of 2.4 using the MIMIR excerpt approach and 2.44 using the Khresmoi summarization technique.

Question	Summarization	Excerpts	First Sentences
'How readable is the summary?'	3.84	2.51	3.99

Table 1: Evaluation results - average rating for the summary generation techniques, for both documents which were relevant and not relevant to the queries, across the five queries.

	Summarization	Excerpts	First Sentences
'How readable is the summary?'	3.89	2.29	4.01
'How relevant is the summary with respect to the query?'	3.83	3.41	3.79

Table 2: Evaluation results - average rating for the summary generation techniques, for documents relevant to the five queries.

Question	Summarization	Excerpts	First Sentences
'How readable is the summary?'	3.79	2.73	3.96
'How relevant is the summary with respect to the query?'	2.44	2.40	1.60

Table 3: Evaluation results - average rating for the summary generation techniques, for documents which were not relevant to the five queries.

6 Conclusions

In this document we described the summarizer developed at Dublin City University (DCU) for the Khresmoi project. The many features of this summarizer were detailed, and its integration point in the Khresmoi prototype shown. First experiments showed promise but also need for further refinement of the summarizer for medical document summarizing.

Given the flexible nature of the summarizer's architecture, it easily lends itself to further summarizer technique development. Ongoing development is exploring new feature development for the summarizer, adapting the summarizer to other languages and adapting the summarizer on medical data used in Khresmoi (medical articles and abstracts). These studies are showing positive results for medical document summarization.

7 References

[1] Khosravi, H., Eslami, E., Kyoomarsi, F., Dehkordy, P.K.: Optimizing text summarization based on fuzzy logic. In Lee, R.Y., Kim, H.K., eds.: Computer and Information Science. Volume 131 of Studies in Computational Intelligence. Springer (2008) 121-130

[2] Alguliev, R.M., Aliguliyev, R.M.: Effective summarization method of text documents. In Skowron, A., Agrawal, R., Luck, M., Yamaguchi, T., Morizet-Mahoudeaux, P., Liu, J., Zhong, N., eds.:Web Intelligence, IEEE Computer Society (2005) 264-271

[3] Bracewell D.B, Ren F, K.S.: Multilingual single document keyword extraction for information retrieval. In: Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, 2005. (2005) 517-522

[4] Gupta, V., Lehal, G.: A survey of text summarization extractive techniques. Journal of Emerging Technologies in Web Intelligence 2(3) (2010)

[5] Lin, C.Y.: Training a selection function for extraction. In: Proceedings of the eighth international conference on Information and knowledge management. CIKM '99, New York, NY, USA, ACM (1999) 55-62

[6] Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research and Development 2(2) (1958) 159-165

[7] Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic text structuring and summarization.

Volume 33., Tarrytown, NY, USA, Pergamon Press, Inc. (March 1997) 193-207

[8] Edmundson, H.P.: New methods in automatic extracting. Journal of the ACM 16(2) (1969) 264-285

[9] Conroy, J.M., O'Leary, D.P.: Text summarization via hidden Markovmodels. In: Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (2001) 406-407

[10] Lin, C.Y.: Training a selection function for extraction. In: Proceedings of the eighth international conference on Information and knowledge management. CIKM '99, New York, NY, USA, ACM (1999) 55-62

[11] Aswani, N., Kelly, L., Greenwood, M., Roberts, A. Samwald, M., Pletneva, N., Jones, G., Goeuriot, L.: D1.3 - Report on results of the WP1 first evaluation phase. Public Technical Report, Khresmoi Project, (2012).