# CROXMLSUM – the System for XML Document Summarization in Croatian

Nives Mikelic Preradovic, Tomislava Lauc, Damir Boras
University of Zagreb, Faculty of Humanities and Social Sciences, Department of Information Sciences, I. Lučića 3
nmikelic@ffzg.hr, tlauc@ffzg.hr , dboras@ffzg.hr

***Abstract*—**The paper describes automatic summarization of the XML documents in Croatian language. The goal of the summarizer is to generate extracts with high percent of extract-worthiness and similarity to the author's abstract. Our research shows that extracts generated using our algorithm are well formed, but it also shows that algorithm is very domain dependant.
The results of the evaluation process proved that the technique of identifying cue phrases and bonus/stigma words in the training corpus significantly improves the text summarization for Croatian language.

The research brought us to conclusion that we should develop the implementation of the Porter's stemming algorithm in order to improve the text summarization for Croatian language, which is currently at an early stage of development.

***Keywords*—**Automatic summarization, XML documents, Croatian language, Perl

## I. Introduction

The goal of automatic summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's needs.

Given a document, text summarization is concerned with the generation of a shorter version which preserves the meaning of the original text.

Automatic text summarization has been under development for many years, and there has recently been much more interests in it due to the increased use of Internet. For example, this technique can be used to summarize news to SMS or WAP-format for mobile phone, or to let a computer synthetically read the summarized text because the written text could be too long and tedious to listen to.

Also, summarization can be used in search engines in order to present compressed descriptions of the search results or in keyword directed news subscriptions of news which are summarized and sent to the user.

Finally, it can be used for searching in foreign languages in order to obtain an automatically translated summary of the automatically summarized text.

The word *summary* is used in a variety of contexts: depending on the input, one can have *single or multiple-document* summaries, while depending on the output one can have *extract* or *abstract-like* summaries.

So, one of the goals of the automatic text summarization is to automatically generate extracts by selecting salient sentences from the original text.

An extract is therefore a summary consisting entirely of the material copied from the input, but the sentences selected are those sentences of a text that are the most representative of pertinent information. The success of such summarization system relies on the use of appropriate features to select the salient sentences.

It is a custom to speak of an extract of n% condensation of input text. More precisely, n% of the input's words may appear in the extract, or n% of the input's sentences may appear in the extract, or even n% of the input paragraphs may appear in the extract.

Extract is therefore a summary extracted from the original text on a statistical basis or by using heuristic methods or a combination of both. The extracted parts are not syntactically or content wise altered, but are the most representative of relevant information.

Abstracts, on the other hand, represent the interpretation of the original text, where the process of producing it involves rewriting the original text in a shorter version by replacing wordy concepts with shorter ones. Implementation of abstract methods requires symbolic world knowledge which is too difficult to acquire on a large enough scale to provide a robust summarization.

Unlike an extract, an *abstract* is a summary which re-phrases content coherently and also contains at least some of the material that is not present in the input. In general, abstracts offer the possibility of higher degrees of condensation: a short abstract may offer more information than a longer extract.

Furthermore, depending on the usage, a summary can be *indicative* or *informative*. An *indicative summary* can provide only an indication of the main topics in the input text. Thus, an indicative abstract aims at helping the user to decide whether to read the information source, or not. By contrast, an *informative* abstract covers all the salient information in the source at some level of detail, i.e., it can reflect to a certain extent the semantic content of the input text as well.

Also, depending on the purpose, a summary can be *generic*. i.e., it can reflect the author's point of view with respect to all important topics in the input text, or it can be *query oriented* (also, *user-focused* or *topic-focused*), in other words, it can reflect only the topics in the input text that are specific to a given query.

Finally, one can distinguish academic approaches to the process of summarization from the commercial ones. The most popular commercial summarizers are Copernic (http://www.copernic.com/index.html), Sinope (http://www.sinope.nl/en/sinope/index.html) and AutoSummarize, embedded as a part of Microsoft Word. Copernic produces summary reports for text contents by processing documents, web pages, hyperlinks, e-mail messages and files. Sinope generates summaries of arbitrary texts, including web pages, by integrating with Microsoft Internet Explorer. AutoSummarize allows summarization of Word documents, but does not allow any structural analysis.

On the other hand, there are academic approaches that try to find the solution to the problem of document summarization from a theoretical point of view as well as practical approaches that focus on the specific applications. Theoretic approaches are distinguished as statistical [13,15], analytical [8,4], information retrieval [1,3] and information fusion [2] approach. Practical approaches relate to baseball program summaries [20], clinical data visualization [18] and web browsing on handheld devices [17].

Our approach to document summarization is an academic approach and the summaries generated by our algorithm are all extracts that aim to give the clue about the main topics in the article, but also to extract the most relevant information from the article.

## II. The system for XML document summarization

Summarization system can have a lot of different parameters, such as: *compression rate* (summary length vs. source length), *audience* (user focused vs. generic), *relation to source* (extract vs. abstract), *function* (indicative vs. informative), *coherence* (coherent vs. incoherent), *span* (single- vs. multi- document summarization), *language* (monolingual, multilingual or cross-lingual), *genre* (special strategies for different varieties of text), *media* (type of media or their combination in input/output) and *linguistics space* that includes dimensions such as level of language processing (morphological, syntactical, etc.) or position (of the word or sentence in the text). In any given application, the importance of these parameters is different and also it is unlikely that any summarizer can handle all of them.

In this paper, extracts can be viewed as a kind of indicative summary which helps readers to judge the relevance of the associated document and decide whether or not the full text is worth reading.

However, extracts which are a collection of sentences from the original text, have a number of drawbacks. The selected sentences lack cohesion when they contain the anaphoric reference or the topic shifts. When a sentence is selected out of the context of its neighbouring sentences, it may be difficult for the readers to determine the references of the anaphora.

Also, a research done by Endres-Niggemeyer [12] show that people prefer extractive summaries instead of abstractive summaries and selection based approach is still the dominant approach in practice.

Many different approaches have been proposed for text summarization.

Luhn [14] utilized word-frequency-based rules to identify sentences for summaries, based on the intuition that the most frequent words represent the most important concepts of the text.

Edmundson [11] incorporated new features such as cue phrases, title/ heading words, and sentence location into the summarization process, in addition to word frequency. The ideas behind these older approaches are still used in modern text extraction research.

In this paper, we describe the summarization of the scientific papers, since in the scientific literature words are mostly unambiguous.

The important words are repeated throughout the text and therefore most of the relevant information should be included in the extract, which is a kind of indicative summary that helps readers to judge the relevance of the associated document and decide whether or not the full text is worth reading.

Although sentences in extracts may lack cohesion if they contain the anaphoric reference or the topic shifts, these problems can be solved through post-processing. Extracts proved to be very useful for people to form an opinion of the context of the original scientific paper.

### *A. Program Overview*

Our summarization system is fairly linear and it consist of ten sections.

The first section checks the program arguments, while the second one extracts sentences from XML file.

In the third section, we implemented the extract generating methodology that includes ranking sentences for a given text by assigning weighted scores based on both the statistical and some linguistic features in the text. In other words, the third section weights each sentence according to the given rules.

The statistical features used were those that are proved to be efficient in the standard monolingual retrieval techniques. So, the approach to text summarization described in this paper allows generic summaries by scoring sentences with respect to both statistical and linguistic features.

This means that extracts are obtained by selecting sentences in the original text [7]. Sentence selection is achieved in two steps. Firstly, each sentence in the text is assigned a score using statistical features to yield a salience function for sentence selection. Secondly, all the sentences are ranked and a predefined number of these top-scoring sentences forms an extract.

Statistical features for weighting sentences used in the first step of the third section include: sentence location and *tfidf* words, while the linguistic features pertain to lemmatization wordform process.

Sentence location feature is based on the hypothesis that sentences occurring after the titles should be relevant and that topic sentences tend to occur at the beginning of a text and a new paragraph.

Therefore, each sentence in the first step of the third section is weighted according to the following location criteria:

- beginning of the paragraph
- end of the paragraph
- title
- heading

Titles and headings are considered to be of infinite weight. As a result they will always be chosen first. Their weight is defined as a large number and that makes impossible for some other sentence to accumulate enough weight to print first.

In the second step of the third section, in order to find and count the *tfidf* words, we collected a list of stop words for Croatian as well as a list of irrelevant document words, taking into account that too low or too high frequency words are mostly insignificant for extracts.

We used Luhn's method [14], which claims that important sentences contain words that occur "somewhat" frequently. Frequency of word occurrence in an article seems to be a useful measurement of word significance so the method increases sentence score for each "somewhat" frequent word.

Basically, a sentence vector is calculated using significant word frequency for all the sentences of the document.

The *significance factor* calculated for every sentence of a document reflects the number of occurrences of significant words within a sentence and the whole document.

The equation (1) calculates the $w_{ik}$ which is the weight of the k-th word in the sentence *i*, calculated by multiplying the word frequency with the IDF, that is log N/n (*N* is the number of sentences in the document, whereas *n* is the number of the sentences the word appears in).

$$w_{ik} = f_{ik} * \log (N/n) \qquad (1)$$

The similarity of the sentences is calculated according to the equation (2) and sentences are ordered in a summary according to their ranking.

$$S(S_i, S_j) = \frac{\sum_{k=1}^{t} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{t} (w_{ik})^2 \bullet \sum_{k=1}^{t} (w_{jk})^2}} \qquad (2)$$

In the equation (2) $S$ is similarity measure, $S_i$ is i-th sentence in the text, and $S_j$ is j-th sentence in the text, $w_{jk}$ is weight of the k-th word in the sentence *j*, $w_{ik}$ is weight of the k-th word in the sentence *i* and *t* is the total number of key words in the text.

The algorithm gives a score to each sentence in a text, using the term count method which is based on the inverse sentence frequency list. It assumes that the subject of the article is the list of ideas that are mostly discussed in the article.

Furthermore, in the fourth section the whole article is scanned once and a hash frequency table of all words is created (i.e. all the words and their occurrence in a text are stored in this table).

After the creation of the hash frequency table, in the fifth section, all stop words and irrelevant words are being removed from the frequency table (i.e. subtracted from the article wordlist).

Stop words and irrelevant words are removed from the table by comparing the list of words in the hash frequency table with the list of stop words.

Stop words in Croatian are grammatical words like prepositions (*u, na o, po*), conjunctions *(i, ali*), some adverbs (*kako, tako*), pronouns (*ja, ti, on*) with all of the case forms (*meni, njoj*), auxiliary verbs (*biti, htjeti*) and modal verbs (*moći, smjeti*) with all of their forms in all verb tenses.

All of the stop words are extracted from the Lexical Database of the Croatian Language [5] automatically.

The list of the irrelevant words (which are in fact words too common for the particular scientific field and because of their high frequency affecting negatively the relevance of a sentence) was generated as well, selecting all of the words in a database assigned as personal nouns and numbers and analyzing them in a semi-automatic way.

In the sixth section, the article wordlist is sorted, and top words (the top percentage of most frequent words that left over) are assumed to be what the article talks about and they are extracted, while the sentences are weighted again according to the given rules.

The idea behind this approach is that the important ideas in a scientific article are described with various but similar words, while redundant information is presented with terms that are less technical and are not related to the main subject of the article. In other words, our hypothesis is that if greatest number of *tfidf* words are found closest together, the probability that the representative information is given is very high.

So, following this main idea, the issue for text summarization in Croatian is whether to apply the sentence scoring with respect to linguistic features (apart from above described statistical features) as well, since the lemmatization process as a factor of the system successfulness still needs to be proved.

In the seventh section, we decided to assign the weighted scores to the words in the article wordlist from the previous section, based on the linguistic features.

Since Croatian is a highly inflected language, there is a problem of counting different wordforms that belong to the same lemma (basic form). In order to determine the new list of *tfidf* words, we used the Lexical Database of the Croatian Language [5] that supports the semi-automatic generation of the basic form (or forms) for a given word from a text and analysis of all the wordforms.

The problem of missing word as well as the word disambiguation problem was resolved manually and by looking up the context of a given word.

Also, some wordforms are additionally lemmatized regarding word formation suffixes (e.g. verbal noun, participle and past participle are assigned to the same basic wordform).

The preliminary text corpus used for producing extract summaries comprises of scientific papers taken from the Croatian Scientific Bibliography database [9], where all the scientific papers include the manually written author's abstracts.

In the eight section, all the sentences (apart from title and header) are weighted once more according to the following rules:

- title word
- header word
- tfidf words - keywords

Title words are given higher values than tfidf and header words, as these are normally what the text is about, similarly for heading words and tfidf words-keywords.

Only a certain percentage of the tfidf words in the text is used as the set of keywords. This percentage is set manually and it depends on the length of the final extract.

TABLE I
SIMILARITY BETWEEN EXTRACTS OBTAINED IN THE EXPERIMENTS AND AUTHOR'S ORIGINAL ABSTRACT

| 1-WORD CUE PHRASES | 2-WORD CUE PHRASES | 3-WORD CUE PHRASES |
|---|---|---|
| predlažemo | utvrđeno je | tema ovog rada |
| smatramo | ovaj članak | u ovom radu |
| zaključujemo | u članku | tema ovog članka |
| | metoda za | u ovom članku |
| | članak je | cilj ovog rada |
| | tema rada /članka | cilj ovog istraživanja |
| | razvili smo | problem istraživanja je |
| | članak /rad ima | ovaj članak opisuje |
| | cilj rada | da pokažemo kako |
| | cilj istraživanja | je da pokaže |
| | eksperimentalni rezultati | u ovoj studiji |
| | preliminarni rezultati | prednosti ovog pristupa |
| | rezultati pokazuju | kontekst ovog članka /rada |
| | fokusirajući se | |
| | problem je | |
| | prethodni rad | |
| | ovog rada / članka | |
| | članak / rad opisuje | |
| | predlažemo metodu | |
| | problem je | |
| | zaključujemo da | |

In the ninth section, sentences are finally weighted according to the cue phrases and stigma/bonus words.

The cue phrases are considered as very useful indicators for locating important sentences in the paper, which contain material central to the theme of the text. Existing methods of identifying cue phrases in English are usually frequency based i.e. high-frequency cue phrases are identified in the original paper.

Cue phrases such as *"this paper presents"* or *"we propose",* which are proved to be important features for text summarization in English [19], are mostly discovered by frequency count, which is used for identifying high-frequency cue phrases.

For the purpose of this experiment, we manually analyzed the abstracts of the scientific papers from the Croatian Scientific Bibliography database and looked for the cue phrases such as "in this paper" and "we conclude" in Croatian. We then compared them to the original author papers and looked for the same cue phrases there. The list of the cue phrases we collected is given in Table 1.

Unfortunately, we discovered that authors tend to use a huge variety of the cue phrases and it is hard to say which of them have a common use along the corpus since the frequency count was very low.

Frequency based methods are, as we realized, not that effective in Croatian, because they can not identify low-frequency cue phrases, which we found to be of high relevance in text summarization for Croatian.

Therefore, we investigated the possibility of exploiting co-occurrence method rather than frequency.

Co-occurrence method is based on the observation that cue phrases and tfidf words - keywords appear in the same sentences.

This combined method [10] definitely outperforms the frequency-based method, which is proved by those experiments on the papers from the Croatian Scientific Bibliography [9].

Also, we looked for the bonus words, such as "Significant" or "Greatest" [16], which positively affect the relevance of a sentence and also for the stigma words, such as "Impossible" or "Hardly", which affect the relevance of a sentence negatively.

Then, if the sentence contains cue phrase and bonus phrase as well, we add a score to the sentence, or we penalize it if it contains a stigma phrase.

Finally, in the tenth section, the top percentage of sentences (as specified by user) are sent to the XML parser that outputs sentences from the original source file to preserve capitalization and punctuation lost in the weighting calculations. This simple parser is by no means perfect, it ignores most tags, but for the preliminary research it achieved fine results.

So, in this program each sentence is basically given a grade based on the word weight in it together with the location criteria and the use of cue phrases and stigma/bonus words. In order to produce 20% summary, 20% sentences with highest grade are extracted and printed. The highest weighing sentences are included in the extract.

### *B. Problems/Difficulties*

The program obviously has limitations. The sentences chosen could be further rectified with more cue phrases or additional stigma/bonus words, which would just require finding them manually.

Although this approach works well, it can be very domain dependant (e.g. scientific research papers differ quite a lot from newspaper articles, so a stigma word relevant to one may not reflect in the other).

Also, deciding on the weighting to give each statistical location rule is difficult and much more testing would produce better values.

After the preliminary testing, the highest weight is assigned to title and headers; it is followed by weight for cue phrases; in the third place are title, header and bonus words which are assigned similar weights, tfidf word is allocated half of the header word weight, while beginning of paragraph and end of paragraph have double weight of the header word.

The length of a sentence seems to be important, as some sentences are obviously longer than other and naturally more likely to have more keywords than some smaller sentence. On the other hand, a long sentence hopefully contains more information than a short one, but not always.

## III. Document pre-processing

The aim of the system for XML Document Summarization in Croatian is to design a summarizer that not only processes traditional "flat" documents, which are primarily textual documents with no structure, but also to process complex structured documents by retaining the structure.

Documents can therefore be categorized into two classes, structured and non-structured.

Structured documents have a well-defined hierarchical structure, such as titles and sections clearly marked with single or multiple level headings. There are also other attributes that create hierarchy, such as distinctive colour, underlines, boldness.

A non-structured document (a "flat" document) has no attributes. This type of document usually has a title, but after that the content is not organized in any structured fashion.

Since XML is a standard textual markup language suitable for encoding almost any sort of data and since it works very well for both unstructured narrative data written by people and for the record-oriented data common in computer applications, we have chosen to design a summarizer for XML documents.

An XML document is made up of nested elements. Each element has a name, a set of attributes and some content. The content can include plain text and/or other elements. The attributes are name value pairs associated with the element. Each document has a single topmost element called the root or document element. Since all non-root elements nest completely inside other elements, an XML document has a natural tree structure. Besides elements and text nodes, XML documents can also contain comments, processing instructions, an XML declaration, and a document type declaration.

Syntactically elements are delimited by tags that look like <Title>, </Title>, and <Title/>. <Title> is a start-tag that must be matched by the corresponding end-tag </Title>. The content of the Title element comes in-between these two tags. <Title/> is an empty-element tag that represents a Title element with no content. Attributes are indicated by name="value" pairs inside start-tags and empty-element tags.

Every XML document must be well-formed. Among other things this means, every start-tag must have a matching end-tag, every attribute value must be quoted, and only certain characters can be used in element names. If a document is not well-formed, it is not an XML document; and XML parsers will not accept it.

All the scientific papers taken from the Croatian Scientific Bibliography [9] were in the Microsoft Word format.

In order to be able to process the structured document instead of the flat one and to test our XML summarizer, we implemented the Visual Basic macro program that removes footnotes and automatically denotes paragraph beginning and paragraph end with the XML tag in the first step of pre-processing. Paragraph mark is replaced with paragraph tag: <P> for beginning of the paragraph and <\P> for paragraph end.

Also, Microsoft Word styles (combination of formatting characteristics, such as font, font size, and indentation, that are automatically named and stored as a set) for Title and Headers (Header 1 to Header 9) were replaced with corresponding XML tag <TITLE>, <HEADERID=1>, <HEADERID=2>…etc.

Furthermore, all sentences are marked with XML tag for sentence. Sentence ID increases for each new sentence.

Beginning of the sentence in our program is defined with the capital letter that follows white space and stop/ question/ exclamation mark or with the manual line break mark that follows stop/question/exclamation mark. (Question mark and exclamation mark can appear at the end of sentence, but it is not so likely in the scientific papers.)

One of the problems is that abbreviations or years can sometimes be misinterpreted as a sentence end.

Therefore, we have to replace all the abbreviations with the full words if the abbreviation does not represent a real sentence end.

Also, if the year ends with stop mark (*e.g.1960.*), the stop mark is removed if it does not represent a real sentence end. Dictionary of abbreviations is taken from the PhD thesis *Theory and rules of automatic text segmentation in Croatian language* [6] and it contains 467 abbreviations.

Although most of the abbreviations were replaced correctly, we still encountered some problems that could be sorted only by further processing (e.g. *dr*. represents both *doktor* and *drugo*).

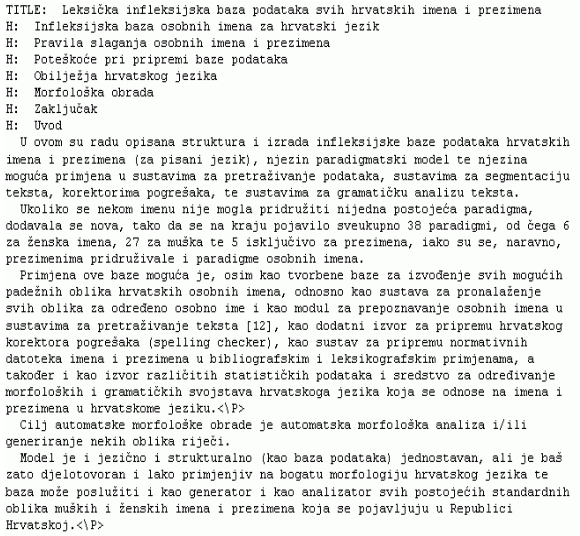
The other problem is errors the authors made in the original paper: double white space before stop mark, inconsistent citations, unsystematic notation of quotations, etc. Unfortunately, this problem can only be solved manually.

After the title, headers, paragraph and sentences are clearly marked with XML tags, another Visual Basic macro program replaces characters *č, ć, đ, ž* and *š* with *cx, cy, dy, zx* and *sx*. This step is performed because Perl module that extracts sentences in summary at this moment does not support Unicode.

After the extracts are acquired, Visual Basic macro program returns the original characters to the text of the extract. Furthermore, the list of the stop words: grammatical words like prepositions (*u, na o, po*), conjunctions (*i, al*i), adverbs (*kako, tako*), pronouns (*ja, ti, on*) with all of the case forms *(meni, njoj*), auxiliary verbs (*biti, htjeti*) and modal verbs *(moći, smjeti*) with all of their forms in all verb tenses was extracted from the Lexical Database of the Croatian Language [6] automatically.

Also, the list of the irrelevant words was generated as well, selecting all of the words in a database assigned as personal nouns and numbers and analyzing them in a semi-automatic way.

Finally, extract summaries were compared to the authors' hand written summaries (summary length in words is narrowly distributed around 150-200 words per summary, or approximately four to six sentences). The results obtained are given in the next section.



```
TITLE:  Leksička infleksijska baza podataka svih hrvatskih imena i prezimena
H:  Infleksijska baza osobnih imena za hrvatski jezik
H:  Pravila slaganja osobnih imena i prezimena
H:  Poteškoće pri pripremi baze podataka
H:  Obilježja hrvatskog jezika
H:  Morfološka obrada
H:  Zaključak
H:  Uvod
  U ovom su radu opisana struktura i izrada infleksijske baze podataka hrvatskih
imena i prezimena (za pisani jezik), njezin paradigmatski model te njezina
moguća primjena u sustavima za pretraživanje podataka, sustavima za segmentaciju
teksta, korektorima pogrešaka, te sustavima za gramatičku analizu teksta.
  Ukoliko se nekom imenu nije mogla pridružiti nijedna postojeća paradigma,
dodavala se nova, tako da se na kraju pojavilo sveukupno 38 paradigmi, od čega 6
za ženska imena, 27 za muška te 5 isključivo za prezimena, iako su se, naravno,
prezimenima pridruživale i paradigme osobnih imena.
  Primjena ove baze moguća je, osim kao tvorbene baze za izvođenje svih mogućih
padežnih oblika hrvatskih osobnih imena, odnosno kao sustava za pronalaženje
svih oblika za određeno osobno ime i kao modul za prepoznavanje osobnih imena u
sustavima za pretraživanje teksta [12], kao dodatni izvor za pripremu hrvatskog
korektora pogrešaka (spelling checker), kao sustav za pripremu normativnih
datoteka imena i prezimena u bibliografskim i leksikografskim primjenama, a
također i kao izvor različitih statističkih podataka i sredstvo za određivanje
morfoloških i gramatičkih svojstava hrvatskoga jezika koja se odnose na imena i
prezimena u hrvatskome jeziku.<\P>
  Cilj automatske morfološke obrade je automatska morfološka analiza i/ili
generiranje nekih oblika riječi.
  Model je i jezično i strukturalno (kao baza podataka) jednostavan, ali je baš
zato djelotovoran i lako primjenjiv na bogatu morfologiju hrvatskog jezika te
baza može poslužiti i kao generator i kao analizator svih postojećih standardnih
oblika muških i ženskih imena i prezimena koja se pojavljuju u Republici
Hrvatskoj.<\P>
```

Fig. 1 example of the 9% extract summary

## IV. Evaluation

Evaluation was done on the corpus of the scientific papers from the Croatian Scientific Bibliography database that were different from the abstracts used for the training of the system.

It was performed through the comparison of the automatically created abstracts from our system with the authors' abstracts taken directly from the Croatian Scientific Bibliography database.

Twelve experiments were performed.

The weight of a sentence in the first experiment is a linear combination of the title, header, location and *tfidf* word-keyword weights, where stop words and irrelevant words are eliminated. The important thing for the *tfidf* word - keyword weights is that both the statistical and linguistic features are taken into account.

The second experiment combines title, location and header weights again, but it also includes bonus and stigma words, while the stop words are not eliminated.

The third experiment combines all four weights from the second experiment, but stop words are excluded as well.

The fourth experiment combines cue words instead of bonus or stigma words with the title, location and header weights. Stop words are excluded.

The fifth experiment combines both bonus/stigma words and cue phrases with the title, location and header weights. Stop words are excluded.

The weight of a sentence in the sixth experiment is obtained using the *tfidf* word- keyword weights, irrelevant words and stop words only. The important thing for the *tfidf* word - keyword weights is that both the statistical and linguistic features are taken into account.

The seventh experiment combines title, location and header weights to calculate the weight of a sentence and it also removes stop words, but no irrelevant words are removed and only statistical features for the *tfidf* word - keyword weights are taken into account.

The eight experiment uses only statistical features for the *tfidf* word - keyword weights and stop words removal to obtain the sentence weight. No irrelevant words are removed.

The ninth experiment uses weights obtained by combining both linguistic and statistical features for the *tfidf* word - keyword weights, title, location and header weights and it also removes stop words to get the sentence weight. No irrelevant words are removed.

The tenth experiment uses only linguistic features for the *tfidf* word - keyword weights and stop words removal to obtain the sentence weight. No irrelevant words are removed.

The eleventh experiment uses only statistical features for the *tfidf* word - keyword weights, irrelevant words and stop words removal to obtain the sentence weight.

The last experiment combines title, location and header weights to calculate the weight of a sentence and it also removes irrelevant words and stop words, but only statistical features for the *tfidf* word - keyword weights are taken into account.

The preliminary results with extracts that differ in length and combination of features were obtained.

The extracts obtained in the first, sixth, eleventh and the last experiment with the same compression ratio are equal. The reason for this is that words that are part of the titles also appear to be the words with the highest *tfidf* and since the compression was very high, the location method has no influence here.

Also, linguistic features for the *tfidf* word - keywords seem to play no role in the sentence weighting, but the list of the irrelevant words appears to be the distinguishing factor.

This is proved by the extracts obtained in the seventh, eight, ninth and tenth experiment, which all appear to be equal.

Furthermore, comparing the extracts of different size and different combination of weights in these twelve experiments, we found out that 2 out of 9 extracts had the best retention ration and were the most similar to the author abstract in the third experiment.

The other seven extracts achieved the best retention ration in the fourth and fifth experiment (actually, results were equal in the 4th and 5th experiment), although 2 out of these 7 extracts had very bad results in third experiment.

Furthermore, two out of nine extracts had a good retention ratio, but not a single sentence from the extract was included in the author's abstract. The reason for this is that sentences in the author's abstract were not contained in the body of the article.

Also, authors were writing abstracts using the phrases such as: *author claims, author describes, author explains*, that are obviously impossible to be found in the body of the article written by that author.

TABLE II
PERCENTAGE OF SIMILARITY BETWEEN EXTRACTS OBTAINED IN THE EXPERIMENTS AND AUTHOR'S ORIGINAL ABSTRACT

| Author's Abstract | Extract |
|---|---|
| Abstract 7 | 78% |
| Abstract 4 | 76% |
| Abstract 5 | 74% |
| Abstract 1 | 63% |
| Abstract 2 | 57% |
| Abstract 9 | 48% |
| Abstract 3 | 43% |
| Abstract 6 | 30% |
| Abstract 8 | 25% |

In order to avoid this situation, it would be wise to give authors clear instructions before they start writing their abstracts.

Hence, we can conclude that all those extracts and abstracts contain the same significant terms and present the most important content, in spite the fact that they consist of different sentences.

Analyzing the obtained extracts, following characteristics were identified:

- Summary length is definitely dependent of document length (summary length of a document that contains 3366 words is 213 words or approximately four sentences where the summary percentage is 9% of the summary. On the other hand, summary length of a document that contains 5478 words is 405 words or approximately 17 sentences.)
- Extract summaries generated using the title, location, header and cue phrase weights (stop words are eliminated) are different from extract summaries that use the same weights, but also the weight for bonus/stigma words
- Extract summaries generated by removing the stop words and irrelevant words and using only the statistical features for the *tfidf* word - keyword weights are different from extract summaries generated on the same base, but without removal of the irrelevant words
- Extract summaries generated using only the linguistic features for the *tfidf* word - keyword weights are not different from extract summaries that are generated using only the statistical features for the *tfidf* word - keyword weights
- Extract summaries where stop words are eliminated differ very much from extract summaries that still contain stop words, actually, they have much higher retention ratio than the latter summaries
- Extracts which summarize the article down to 1-5% appear to be too short to be compared to the authors' abstracts (their size is half of the size of the author's abstract on average), although the extract worthiness is quite high because they consist of title and headers

Results are expressed as the document vector similarity between an abstract and an extract document.

Nearness between author's abstract and extract obtained by our summarizer is calculated in three steps: list of words is created for each document, word lists are merged together, stop words are removed, the lemmatization is performed and words that are common are extracted in the output.

Finally, nearness (document similarity) is expressed as the fraction, which has *number of word common to document pairs* as numerator and *number of words in the shorter of the two documents in the pair* as denominator.

The interesting fact is that although 2% extracts obtained by the algorithm that removes stop words and irrelevant words contain different sentences than the 2% extracts obtained by the algorithm that removes stop words only, they both achieved almost the same percentage of similarity with the authors' abstracts.

Hence, we can conclude that those extracts contain the same significant terms and present the most important content, in spite the fact that they consist of different sentences.

The similarity percentage between extracts and abstracts are presented in Table 2.

Another evaluation criterion was extract-worthiness or retention ratio.

Results obtained for both the high percentage extracts (10-15%) and the low percentage extracts (5-8%) show that over 90% of the sentences selected are extract-worthy; in other words, extract-worthiness does not get lower with the higher compression ratio, as one may suspect.

Also, 1% and 2% extracts contain only the title of the document, while extracts in the range of 2-5% include title and headers as well.

## V. CONCLUSION

One of the goals of the automatic text summarization is to automatically generate extracts by selecting salient sentences from the original text.

An extract is therefore a summary consisting entirely of the material copied from the input, but the sentences selected are those sentences of a text that are the most representative of pertinent information.

The aim of the CROXMLSUM – system for XML Document Summarization in Croatian is to take an scientific paper in Croatian as the input and to generate an extract with the high retention ratio and about the same size as the author's abstract, if the author's original abstract is available for comparison.

Extracts are obtained by selecting sentences in the original text. Sentence selection is achieved in two steps.

Firstly, each sentence in the text is assigned a score using some features to yield a salience function for sentence selection. Secondly, sentences are ordered in a summary according to their ranking and a predefined number of highest weighing sentences are included in the extract.

We believe that the key facts in a scientific paper are expressed with a range of related words, while redundant information is presented with terms that are not related to the main subject given in the title or the header of the article.

Also, the cue phrases are considered as very useful indicators for locating important sentences which contain material central to the theme of the text.

Unfortunately, we discovered that the authors in Croatian tend to use a huge variety of the cue phrases and it is hard to say which of them have common use along the corpus. Frequency based methods are therefore not effective in Croatian, because they can not identify low-frequency cue phrases, which we found to be of high relevance in the text summarization for Croatian.

Apart from cue phrases, we also looked for the bonus words which positively affect the relevance of a sentence and for the stigma words which affect the relevance of a sentence negatively. Then, if the sentence contained cue phrase and bonus phrase as well, we added a score to the sentence, or we penalized it if it contains a stigma phrase.

We found out that the summary length is definitely dependent of document length.

We also found out that extract summaries where stop words are eliminated differ very much from extract summaries that still contain stop words, actually, they have much higher retention ratio than the latter summaries.

Also, the extract summaries generated using the title, location, header and cue phrase weights (stop words are eliminated) are different from extract summaries that use the same weights, but also the weight for bonus/stigma words, but the retention ratio is not that different .

That means that the system does perform better when using both the cue phrases or bonus/stigma words, but not significantly better.

Finally, extracts which summarize the article down to 1-5% still appear to have the quite high worthiness.

Since the cue phrases proved to be important part of the summarization process, the aim of the future work is to consider the development of the automatic techniques for identifying cue phrases from training corpus for the purpose of the text summarization.

In order to improve our system we plan to keep track of some visual presentation details in the original article, such as font size of words. Words in a larger or bolder font could be weighted higher than other words.

Also, since we concluded that the lemmatization process does not influence the system performance significantly, some in depth linguistic approach to lemmatization should be considered and implemented or some other linguistic features should be taken into account.

Therefore, in order to improve the current system, we plan to implement the Porter’s stemming algorithm for Croatian language.

Up to this point, we have tested the Perl implementation of the Porter's algorithm for English language.

The Porter stemming algorithm (or ‘Porter stemmer’) is a process of removing the common morphological and inflexional endings from words in English. The algorithm was originally described in Porter M.F, 1980. *An algorithm for suffix stripping*, Program, 14(3):130-137.

We downloaded the Perl version of the algorithm from the official home page for distribution of the Porter Stemming Algorithm (http://www.tartarus.org/~martin/PorterStemmer) and implemented it into our summarizer.

We tested the Porter stemming algorithm on the English version of one article from the Croatian Scientific Bibliography database [9]. The extract obtained for English encourages us to implement the same algorithm for Croatian as well.

Finally, if the application of the Porter stemmer shows no improvement in the summarization, we are planning to use some linguistic technique that takes into account the contacts between the sentences.

Most probably, we will use the average lexical connectivity: method that compares the number of terms shared with other sentences. The underlying assumption is that a sentence sharing more terms with other sentences is more important. Those experiments will also be performed on the Croatian Scientific Bibliography.

## References

[1] Aho A., Chang S., McKeown K., Radev D., Smith J., Zaman K. Columbia Digital News Project: An Environment for Briefing and Search over Multimedia. International Journal on Digital Libraries, 1(4):377-385. 1997.

[2] Barzilay R., McKeown K., Elhadad M. Information fusion in the context of multi-document summarization. In Proceedings of ACL'99, 1999.

[3] Berger A., Mittal V. Query-relevant summarization using FAQs. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. 2000.

[4] Boguraev B. Neff M. Discourse Segmentation in Aid of Document Summarization. In Proceedings of Hawaii International Conference on System Sciences (HICSS-33), Minitrack on Digital Documents Understanding, IEEE. 2000.

[5] Boras D. Rječnička baza kao osnova za izradu automatskog detektora pogrešaka teksta na hrvatskom jeziku pisanog pomoću kompjutora. In: Tkalac S, Tuđman M, editors. Informacijske znanosti i znanje. Zagreb: Zavod za informacijske studije; 1990. p. 57-73.

[6] Boras D. Teorija i pravila segmentacije teksta na hrvatskom jeziku. PhD thesis. Zagreb, 1998.

[7] Brandow R., Mitze K., Raul LF. The Automatic Condensation of Electronic Publications by Sentence Selection in Information Processing and Management 1995; 31(5): 675-685.

[8] Brunn M., Chali Y., Pinchak C. Text Summarization Using Lexical Chains. Work on Text Summarization. 2001.

[9] Croatian Scientific Bibliography. URL: http://bib.irb.hr/index.html?lang=EN [1/21/2008]

[10] Daille B. Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. In: Klavans J, Resnik P, editors. The Balancing Act: Combining Symbolic and Statistical Approaches to Language. Cambridge: MIT Press; 1996. p. 49–66.

[11] Edmundson HP., Wyllys, RE. Automatic Abstracting and Indexing - Survey and Recommendations. In: Communications of the ACM 1961; 4(5): 226-234.

[12] Endres-Niggemeyer B. Human-Style Www Summarization; 2000. URL: http://transfer.ik.fh-hannover.de/ik/person/ben/Human-styleSummaNew.pdf [1/21/2008]

[13] Knight K., Marcu D. Statistics-Based Summarization - Step One: Sentence Compression. AAAI/IAAI, p. 703-710, 2000.

[14] Luhn H.P. The Automatic Creation of Literature Abstracts in IBM Journal of Research & Development 1958; 2 (2): 159-165.

[15] McKeown K., Barzilay R., Evans D., Hatzivassiloglou V., Kan M., Schiffman B., Teufel S. Columbia MultiDocument Summarization: Approach and Evaluation. Workshop on Text Summarization, 2001.

[16] Paice CD., Jones PA. The Identification of Important Concepts in Highly Structured Technical Papers. In: Proceedings of the 16th International Conference on Research and Development in Information Retrieval (SIGIR'93); 1993. p. 69–78.

[17] Rahman A, Alam H., Hartono R., Ariyoshi K. Automatic Summarization of Web Content to Smaller Display Devices. 6th International Conference on Document Analysis and Recognition, ICDAR01, p. 1064-1068, 2001.

[18] Shahar Y., Cheng C. Knowledge-based Visualization of Time Oriented Clinical Data. Proceedings of AMIA Annual Fall Symposium p. 155-9, 1998.

[19] Teufel S., Moens M. Sentence Extraction and Rhetorical Classification for Flexible Abstracts; 1998. URL:http://www.cl.cam.ac.uk/users/sht25/papers/aaai98.pdf [1/21/2008]

[20] Yong Rui Y., Gupta A., Acero A. Automatically extracting highlights for TV Baseball programs. ACM Multimedia, p. 105-115, 2000.