Computer Science • Vol. 7 • 2005

Emilia Branny*, Marek Gajęcki**

TEXT SUMMARIZING IN POLISH

The aim of this article is to describe an existing implementation of a text summarizer for Polish, to analyze the results and propose the possibilities of further development. The problem of text summarizing has been already addressed by science but until now there has been no implementation designed for Polish. The implemented algorithm is based on existing developments in the field but it also includes some improvements. It has been optimized for newspaper texts ranging from approx. 10 to 50 sentences. Evaluation has shown that it works better than known generic summarization tools when applied to Polish.

Keywords: *natural language processing, text summarizing*

STRESZCZANIE TEKSTU W JĘZYKU POLSKIM

Celem artykułu jest zaprezentowanie algorytmu streszczającego teksty w języku polskim. Mimo istnienia algorytmów streszczających teksty, brak jest algorytmów dedykowanych dla języka polskiego. Przedstawiony algorytm bazuje na istniejących algorytmach streszczania tekstu, ale zawiera kilka ulepszeń. Algorytm jest przeznaczony dla streszczania tekstów prasowych liczących od 10 do 50 zdań. Przeprowadzone testy pokazują, że algorytm działa lepiej od znanych algorytmów zastosowanych dla języka polskiego.

Słowa kluczowe: *przetwarzanie języka naturalnego, streszczanie tekstu*

1. Introduction

The idea of text summarization is not new, and the concept of a summary is well grounded in everyday experience. To summarize is "to make into a short statement of the main points", as defined by Merriam & Webster Online Thesaurus [1]. This simple explanation can be elaborated within a more scientific approach. According to Van Dijk [2], summarization is the result of a process which consists of three phases:

1. Translation of a text from natural language into a logical model.
2. Macrooperations within this model, which allow to discover the macropropositions – the most important statements and macrostructures, the organization principle of ideas and arguments in the basic text.
3. Translating the macropropositions into a summary – a text in natural language.

*PhD Student EAIiE, AGH-UST, Kraków, Poland
**Institute of Computer Science, AGH-UST, Kraków, Poland, mag@agh.edu.pl

However, in Information Technology understanding and producing natural language is still a big problem. It is not possible to simulate the phases of summarization, which involve producing a logical model from a text, performing generalizations and deletions on the resulting propositions and then generating a summary in natural language. It is because this method, called summarization by abstraction, would require:

- a parser which is able to detect the structure of the sentence;
- sentence surface dependency solver dealing with such issues as:
  - pronoun dependencies,
  - the issue of co-reference,
  - relations of time and place,
  - relations of cause, purpose, condition, concession and other represented by junctions,
  - lexical relations between concepts (exclusion, inclusion, intersection),
  - presupposition (a sentence can convey a presupposition that something else is true, e.g. P: "Peters dog is sick" carries a presupposition that Peter has a dog; the presupposition is true also if $\neg P$ ("Peters dog is not sick"),
  - entailment (if a sentence implies something, but its negation does not e.g. Karen is married to John entails "John is married to Karen"),
  - the relation between the topic and comment (thema and rhema, already given part of the sentence and the part containing new information);
- a language interpreter able to operate on meaning, especially to perform abstraction, to detect meaning relations between lexical units (inclusion, exclusion, etc.), and to account for complex relations between ideas (consequence, alternative, equivalence, etc.);
- a language generator able to generate correct utterances in a natural language, which would convey a given meaning.

As we have mentioned before, current state of art does not allow to build a text summarizer based on this method, especially for Polish, where the required tools are far from available. A less powerful (but also less costly) alternative to abstraction is the method of extraction. Its main advantage is that it does not require "understanding" the meaning of the original text by the machine. The problem with this method is the constraint that it imposes on the summary: it has to consist only of phrases coming from the original text. It is easy to imagine a text, which has so many interdependencies between the sentences, that making a good summary by the method of extraction is impossible. In this situation the resulting summary may not conform to textual standards and it may not reflect the structure of the original text in a correct way. The conclusion is that the method of extraction should be expected to provide no more than an approximation of a correct solution of the problem of summarization.

The method of extraction involves the following steps:

1. Dividing the original text into basic processing units (usually sentences) and marking the beginning and the ending of each unit.

2. Evaluating each unit by multiple criteria.
3. Producing the summary by rewriting these units which have been assigned the highest score.

The most difficult step is the second one evaluation of text units. A number of methods have been proposed, an extensive overview of which can be found in FarsiSum [5] and GreekSum [6]. Here we shall recall the comprehensive list of criteria provided by Lin [7], which encompasses most of the earlier findings, and we shall briefly signalize the directions of current development.

The criteria suggested by Lin [7] are the following (the list and explanations as provided by Mazdak [5] and Pachantouris [6]):

1. **Baseline:** This is a scoring system according to which sentences take their marks depending on their place in the text. For newspaper texts, the first sentence of the text gets the highest ranking, while the last get the lowest.
2. **First sentence:** Similarly to the previous condition, the first sentence of each paragraph of the text is considered to be very important.
3. **Title:** The words included in the title along with the following sentences get a high score.
4. **Word Frequency:** Words, called open class words, which are frequent in the text, are more important than less frequent. The sentences including such keywords that are most often used in the passage usually represent the topic of it.
5. **Indicative Phrases:** Sentences containing phrases like "...this document...".
6. **Position Score:** There is a theory that certain types of documents have their key meaning in certain parts of it. For example in the newspaper text, the first four paragraphs are the most important, while in technical papers the conclusion section is the most important part.
7. **Sentence Length:** The score given to a sentence reflects the number of words in a sentence, normalized by the length of the longest text in the passage.
8. **Proper Name:** Sentences which contain proper nouns get a higher scoring.
9. **Average Lexical Connectivity:** The sentences that share more terms with other sentences are scored higher.
10. **Numerical Data:** The sentences that contain any sort of numerical data are scored higher than those that do not contain.
11. **Proper Name:** Certain types of nouns, like people's names, cities, places etc are important in newspaper texts and sentences containing them are scored higher.
12. **Pronoun:** Sentences containing a pronoun (reflecting co-reference connectivity) are scored higher than those that do not contain.
13. **Weekdays and Months:** Sentences containing names of weekdays or months are scored higher.
14. **Quotation:** Sentences containing quotations may be important for some sort of questions, input by the user.

15. **Query signature:** When a user requires a summary he or she usually has a certain topic on his/her mind. The query of the user affects the summary in that the extracted text will be compelled to contain these words. Normalized score is given to sentences depending on the number of query words they contain.

The list provided by Lin is based on earlier works, in particular:

- Luhn [8], who introduced word-frequency-based rules to identify sentences to use in summary.
- Edmundson [9] who added cue phrases, the presence of title/ heading words and sentence location as evaluation criteria.

The list was the basis for the Scandinavian text summarizer ScandSum [3, 4] and its versions ported to other languages. For example ScandSum uses the following criteria (list and explanations as provided by Mazdak [5] and Pachantouris [6]):

1. **First line:** It should always be included in the summary. This is done by assigning it a very high score, 1000 in SweSum [11].
2. **Position:** Because SweSum supports two kinds of text, Newspaper and Report, the position score depends on the text to be summarized. In newspaper text the first line is the most important and gets the highest score, followed by the others. There is a mathematical formula being used for defining the position score: Position $score = (1/linenumber) * 10$.
3. **Numerical values:** Whenever a number is identified in a text, the line that includes it gets one additional point.
4. **Bold Text:** By identifying the ⟨B⟩ symbol in the HTML code, SweSum assigns the score 100 to lines containing bold text. This is because it sometimes shows the beginning of a paragraph or the first sentence of it.
5. **Keywords:** They are automatically identified as the most frequent words on the passage. The sentences that contain more keywords take a higher score than those that contain fewer or none.
6. **User Keywords:** The play similar role as the keywords described above, but the user defines which words should be used as keywords.

For each sentence the scores resulting from individual criteria are multiplied by predefined language-dependent coefficients and added together to calculate the score of each sentence:

$$Sentence\ score = \sum (C_j * P_j)$$

where:

C_j — predefined language-dependent coefficient,

P_j — score for j-th criterium.

Afterwards, 30% of the sentences (those with the highest scores) are rewritten to the output.

2. Defining the Problem

The main defining features of a summary, which are commonly agreed on, are the following:

- it is a text,
- it is an account of some "original" text,
- it presents the main points of the original text,
- it is supposed to be shorter and more concise than the original.

The problem of summarization has several important characteristics, which have to be taken into account in defining and evaluating a valid summary:

1. It is a heuristic problem, which means that it has more than one correct solution. This results from the nature of language, well expressed by Chomskian transformation grammar. The same meaning can be expressed in various ways, e.g. "She is not married" is equivalent to "She is single", but also "We were not able to buy the tickets" can be equivalent to "The tickets were sold out" in certain contexts.
2. The correctness of a solution is arbitrary, because the criteria are fuzzy.
3. The solution can be approximated, which means that a summary which fulfils the criteria up to a certain degree can also be accepted as valid, although it will be rated lower than the summary which fulfils the criteria better (contains fewer errors, gives a fuller account of the original text etc.)

It appears that the evaluation criteria of a summary should refer to two kinds of features:

1. Relation to the original text:
    a. Does the summary give correct information about the possible world described by the original text?
    b. Is the summary a valid representation of the original (does it give correct impression about the original itself, its scope, structure and content?).
2. Relation to other textual standards, in particular:
    a. Is the summary coherent?
    b. Is the summary cohesive?
    c. Is the summary informative enough?
    d. Does the summary meet the needs of the reader?

These criteria define a correct summary. However, as it has been stated above, although we have formulated them as "yes-no" questions, in fact they are fuzzy and they can be measured quantatively. If some of them are not fulfilled in 100% the summary will of course not be correct in every aspect but it may still be an acceptable and valuable. The purpose of the summarizing algorithm would be to generate summaries of a relatively good quality.

3. The algorithm

The ST algorithm works in the following steps:

1. Divide the original text into sentences; mark the beginning and the ending of each sentence and each paragraph.
2. Generate frequency lists of nouns, numerals and proper names for the original text as a whole and for individual sentences.
3. Evaluate each sentence on the basis of multiple criteria and producing the summary by rewriting these sentences which have been assigned the highest score.

   Beneath we will discuss the steps of the algorithm in detail.

3.1. Step one

The aim of step one is to divide the text into sentences and paragraphs.

The text is divided into sentences and paragraphs by means of a separator "|" (marking sentence boundaries) and "‖" (marking every new line) which have been chosen because they usually do not appear in texts. A sentence is defined as the smallest part of text between two marks from the set: "." "!" "?".

The exceptions are the periods which:

- belong to popular abbreviations such as "tj.", "dot.", "itp." (the list of abbreviations is retrieved from a manually created file);
- follow a single letter, as in abbreviated names ("A. Nelson") and one-letter abbreviations ("a.");
- follow a number, as in "33. Mistrzostwa Polski", "4. miejscu".

The last rule causes a mistake when a sentence ends in a number, because then it will not be separated from the following sentence. However, taking two sentences for one is a better choice than cutting a sentence into two pieces, which would usually ruin the sense of the sentence.

Higher accuracy could be achieved here in the future by:

- using a list of most typical contexts where numbers appear inside sentences (semantic criterias such as: occurrence of a proper name after the period; the text being about sports events etc.) and treating all the other occurrences of period after a number as a sentence boundary. This could increase the accuracy a little bit but the cost of a mistake would still be high (the sentence would get divided in the middle);
- using a parser to find out which solution is correct. This would be the best solution but unfortunately there is no Polish parser available.

The algorithm marks as a paragraph:

- the smallest piece of text between two newline characters,
- the smallest piece of text between the beginning of a text and the first newline character,

- the smallest piece of text between the last newline character and the end of the text.

This gives accurate results in most cases, but there are cases in which the markup is not exactly correct because the unit should be classified as a special element:

- titles, heading,
- list elements,
- signatures,
- photograph captions.

The units mentioned have special function, when compared to a paragraph, which marks the beginning of a new idea. A heading or title marks a topic of the following section. A list element belongs to one unit together with all other elements of the same list. A signature is information about the author of the text, so it does not carry structural information (it is metainformation about the text). A photograph caption represents or refers to a photograph, which is usually not present in plain text.

The enumeration is provided here just to signal the fact that a paragraph as marked in step one of the algorithm is not identical with the paragraph as a structural unit of the text. Some of the distinctions between the units mentioned here are made later by the algorithm (see: headings and lists markup). But the distinctions that do not affect the output (signatures and captions) are never made in this summarizing algorithm.

3.2. Step two

The purpose of this step is to find out about nouns, proper names and numerals in the text.

3.2.1. Nouns

The idea behind finding nouns is that they represent the topics of the text as well as the topics of individual sentences. That is why it is important to identify:

- which nouns appear in the beginning of the text (these are believed to be the most important ideas in the text),
- which nouns appear in the text several times (the sentences which contain these are believed to be more representative for the text and more closely connected to other sentences than the sentences that contain only unique nouns).

Noun frequency lists are needed for each sentence as well as for the whole text. A lexical scanner with a noun filter is needed to tokenize nouns because otherwise it would not be possible to account for multiple occurrences of the same noun (which is the core of this approach to topics). The algorithm uses the ILP library [14], which provides an interface to Polish Inflection Dictionary. The text is scanned with a simple lexical scanner which uses ILP to check the category and base form of the words. The scanner identifies all the nouns in the text and performs tokenizing (assigns to them the ID of their base form).

The problems with noun statistics are that:

- ILP library has no disambiguation and it returns all matching IDs.
- There is no lexicon of synonyms or other mechanisms for finding different words with the same meaning or reference.

As regards disambiguation, in step three the algorithm will choose one ID for each word form from the list of IDs retrieved by the lexical scanner. The ID will be chosen which is the most common in the given text. We can assume that in most cases even if a word has several meanings (e.g. "zamek" – a castle, a zip, a lock) only one of these meanings is present in a short newspaper text. Exact lexical meaning is of course not important as long as we can answer the question where else in the text this noun can be found.

3.2.2. Proper names

The idea behind finding proper names in the text is twofold:

1) proper names represent the topics in the text
2) proper names often provide valuable and precise information to the reader

Therefore a solution must be proposed, which accounts for multiple occurrences of the same proper name. There is no lexical scanner for proper names and it is impossible to devise one, because proper names are practically impossible to gather in a dictionary. That is why identification has to be based on stems rather than dictionary (in Polish proper names are inflected so their form and length can be different in different places). The stem is obtained from a noun by means of a simple computation:

- if the word consist of capital letters only, the stem is the whole word (this allows for necessary precision in abbreviations),
- in other cases the stem is the first half of the word plus so many following letters, that the ending that is cut off is shorter than 5 letters.

However, the algorithm does not produce a stem list. It compares two words and if the stem of the longer word form is found in the beginning of the shorter word form, the shorter word form is treated as an inflected occurrence of the longer one. Let us call it stem-based comparison.

Why is the longer form left? It is because without inflection rules we cannot determine whether a form has an inflection ending or whether it is in the nominative. The longer form is usually an inflected one, so the stem will be more accurate.

Let us now explain from the beginning how proper names can be identified in a text. ST treats as a proper name unit every word spelt with capital letter, not standing in the beginning of a sentence. A proper name unit can be a proper name itself or part of a compound proper name. After a series of stem-based comparisons between proper name units found in the text we are able to:

1. Find their occurrences in the beginnings of the sentences (by stem-based comparison).
2. Detect compound proper names.

The detection of compound proper names is a very complex problem and there is one inevitable source of inaccuracy: the algorithm works on an individual text only, and without any inflection patterns which could detect lack of concordance between units. Therefore it cannot recognize that a phrase should be separate if it is separate nowhere in the text, e.g "stolica Gwatemali Izapa" or "Krakowa Markiem Nawarą". The implemented algorithm works by comparing occurrence lists of individual proper name units. It takes into account several cases, in which it tokenizes the neighboring units as one proper name:

- Several units are always used together in the text and none of them is a part of any other proper name in this text.
- A single unit is always used with certain other unit and never is used separately. The unit is then treated as an optional part of the compound proper name (e.g. in a name, where the surname is obligatory and the name is sometimes omitted).
- A single unit is always used with one of several other units and never separately. The unit is then treated as a dependent part of several compound names and tokenized together with the distinctive unit of these names. (e.g. in "Bank Handlowy", "Bank Spółdzielczy". "Bank" is redundant for identification here, because it is always together with some other part).

Afterwards the proper names are tokenized and a frequency list is build for all the sentences separately and for the text as a whole. Compound proper names are of course treated as one item (they get one and the same ID).

3.2.3. Numerical data

The idea behind finding numerical data in the text is that the sentences containing numbers and numerals are more valuable (informative) to the reader than those which do not contain numbers or numerals. There is no need to identify the numerals/numbers denoting the same number in a text. It is sufficient if their presence is marked. The numerals/numbers are divided into three groups which reflect their importance:

1) Arabic numbers,
2) Roman numbers,
3) Numerals (a number expressed as a word).

Their occurrences are counted separately for each sentence and the types mentioned are scored differently (Arabic numbers are multiplied by the highest coefficient, numerals by the lowest one).

Arabic and Roman numbers are easy to detect with the help of regular expressions and this is done in step three. However, the detection of numerals is partly based on ILP. The lexical scanner which identifies nouns has also an extra filter for numerals. The data retrieved in this way is placed together with data about nouns in .se file. At the end of each line in this file there are numerals and the last position before the newline character is the number of numerals. The number of numerals tells how many positions in the line are occupied by numerals and how many by noun ids.

The lexical scanner in ILP is currently not able to detect all the numerals. That is why scanning for numerals is completed in step three by means of regular expressions.

3.3. Step three

In step three, sentence scoring is performed. It is not a simple combination function which defines the score for each sentence, because the scores are not always mutually independent.

The basic score of every sentence depends on the following characteristics, which are mutually independent:

1. **Relevance to the main topic of the article.** This is measured by the presence of topic words – nouns taken from the first paragraph and from the first sentence of the second paragraph. For every "topic word" present, a sentence gets $P1$ points.
2. **Position of the sentence in the paragraph.** The first sentence in the paragraph always gets $P2$ points.
3. **Presence of proper names.** For each proper name a sentence gets $P6$ points multiplied by the number of occurrences of this proper name in the text.

The basic score of every sentence is also independent of the folowing characteristics which are dependent on other characteristics:

1. **Baseline/Presence of numerical data.** If a sentence has some keywords (nouns that are not unique) then for each Arabic number the sentence gets $P3$ points, for each Roman number – $P4$ points, whereas for each numeral – $P5$ points.
2. **Baseline/position of the sentence (only after a sentence which has topic nouns and begins a new paragraph).** If the sentence is first in the paragraph and if it has some keywords (nouns that are not unique), it should get extra score $P7$. If the sentence is after a sentence which has a positive baseline score, it should get extra score $P8$. This mechanism is called forward conditional score propagation.
3. **Position of the sentence before a paragraph which has higher score.** If a sentence inside the paragraph has a higher score than preceding sentences in the same paragraphs, $1/P9$ of its weight is propagated back to those sentences – added to their score. This mechanism is called backward score propagation inside a paragraph. It is performed as the last step, after score differentiation which is described below.

In the current version of the algorithm the values of the parameters are the following:

$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$	$P8$	$P9$
4	3	4	1	2	0.5	4	1	10

The desired compression rate is 30%, therefore the text is reduced to that size. The sentences with the highest scores are rewritten to the summary. The rest is omitted. In order to avoid the situation where many sentences have the same score and for that reason the expected length of the summary cannot be achieved, score differentiation has been introduced. Score differentiation is an operation of multiplying the scores of all sentences by:

$$y = ax^2 + bx + c$$

where x is the sentence number, and a, b, c are coefficients such that:

$$c = 1.1 \qquad b = -0.2/minimum \qquad a = -b/(2 * minimum).$$

The minimum of the function is situated at x:

$$x = (2 * L)/3$$

where L is the total number of sentences in the text.

The result is that the score of the sentence in the beginning of the text gets multiplied by 1.1, the score of the sentence in the minimum gets multiplied by 1. The new score S is obtained from the previous score $S0$ by:

$$S = S0 * y.$$

However, this "standard" compression rate does not seem to be always a good idea. There should be other criteria connected with text structure and not imposed from the outside. There should be also a configurable maximum length (which could be considerably shorter than $1/3 * L$ in case of long texts). Some text get summarized better or the user needs them shorter, but some text cannot be summarized, e.g. a telephone book or a recipe.

4. Examples

In appendix A we will present a examples of summarie. In these examples the sentences chosen for the summary are in bold. The rest of the text is in standard font. The sample of text have been produced from "Rzeczpospolita" corpus.

5. Evaluation

The evaluation method used was especially designed to take into account the criteria mentioned in the section "Defining The Problem". Among existing methods [10] there is no method which would take into account our criteria. This is the case for example with a popular method of "gold standard" which compares the set of sentences chosen by people and the set of sentences chosen by computer. It does not test coherence or informative content of the summary.

We are not going to discuss the evaluation method here in detail because it is complex and it deserves a separate study. We will just say that it is based on text linguistics and on input from users.

It takes into account three criteria:

1) informative content,
2) misinformation,
3) T-grammaticality (which stands for correct relationships between sentences of a text).

Here we will only present the results of evaluation. The Figure 1 shows performance of different summarizing algorithms based on extraction. The detailed performance test was effectuated on three Polish texts from "Rzeczpospolita" corpus on which the ST algorithm had not been tested during development.

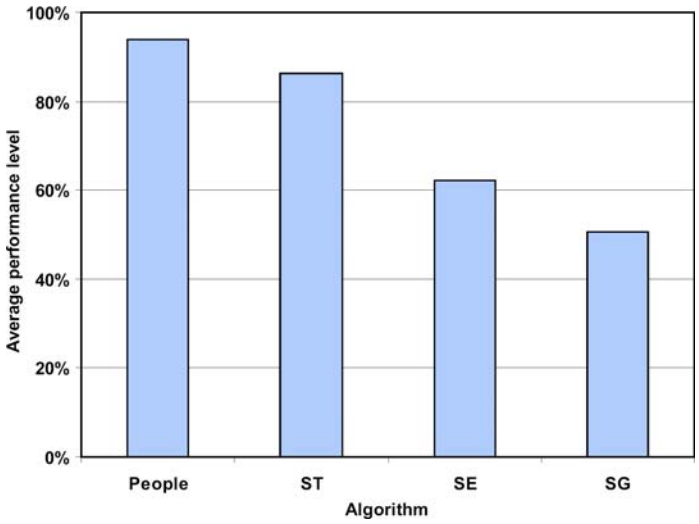

Fig. 1. Performance of summarizing algorithms based on extraction

The score of 100% indicates a correct summary as defined in "Defining the Problem" section.

People – average result for people who were asked to make extraction summaries manually, within the constraints formulated for the algorithm (1/3 of the original amount of sentences rewritten to the output).

ST result by ST algorithm (algorithm described in this article).

SG result by SweSum Generic with default settings for newspaper texts.

SE result by SweSum[11] on English translation of the Polish texts.

Conclusions which can be drawn from the comparison are the following:

- Algorithms designed for a specific language work better than language-independent versions based on the same principles.
- Sometimes even humans are unable to generate a 100% correct summary by extraction within certain space limits. It may suggest that we should resign from imposing a rigid constraint on the length of the summary.

- A paragraph-based sentence scoring, using only nouns in word statistics and other techniques used in the algorithm seem to be an improvement in the summary-making technology.

However, the text corpus used for evaluating summaries is currently too small to prove that the conclusions are generally true outside the specific domain of texts. It would also be beneficial to compare more algorithms.

The following Figure 2 shows the overlap between sentences chosen by the computer and the people in different cases. 100% means the total of sentences in a summary. The chart shows how much the output from the algorithms was overlapping with the answers by people and also how much the output from the people was overlapping with other peoples answers. It is an average from three texts.

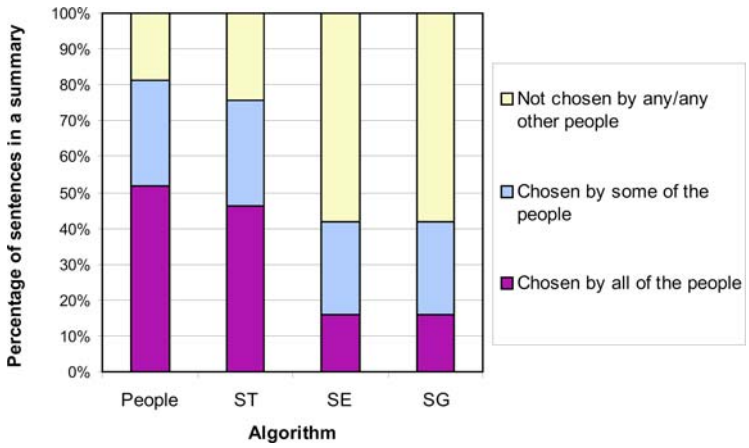

Fig. 2. Sentences chosen

Conclusions which can be drawn:

- It is true that the summaries created by people do not overlap entirely [12].
- It was surprising that even for a couple of people (usually 3–4 people per text) the number of sentences which overlapped in all answers was 50%.
- The proportion of overlapping sentences does not reflect the differences in informative content and T-grammaticality of summaries which can be detected in the complex evaluation method. It can be seen that SE and SG would have the same relation to the "golden standard" although their performance differs considerably.

The main conclusion is that although the "golden standard" method and counting overlapping sentences can give an idea about the performance of the summary, in fact it fails to track many real issues which make one summarization method work better than the other. It is therefore not good enough for rating summarizing algorithms against each other.

6. Conclusions

Conclusions from this work refer to different issues ranging from technical developments and ways of building a summarizing algorithm to a revision of existing evaluation methods. For detailed conclusions it is better to visit the relevant chapters of this work. Here we will just outline the most important conclusions referring to major problems of text summarization.

Is there a need for a Polish text summarizer? The summarizing algorithm which has been built especially for Polish and uses Polish inflection dictionary works considerably better than a tested generic tool. In this case language-specific tools demonstrate their superiority over language-independent text summarizers.

What can be improved in text summarizing algorithms? The algorithm represents a certain approach to summarizing. It emphasizes the informative value of the summary and also its grammatical and logical acceptability. That is why it used a number of specific solutions:

- paragraph-based sentence scoring (vs paragraph-independent scoring),
- using only nouns in basic word statistics (vs all-inclusive statistics),
- using the idea of key words coming from the beginning of the article to rate sentences (vs using title words or position-independent scoring),
- influencing sentence score by the score of adjacent sentences,
- recognizing the end of sentence with the help of abbreviation dictionary.

The test for these solutions was a comparison with human results and the results of other tested language-dependent and language-independent algorithms. The tests have proved that the solutions mentioned are leading in a good direction and they have the potential to improve the performance of automatic summarizing.

How to implement proper names recognition for text? The tests have shown that for the purpose of summarizing we do not need a 100% accurate proper names recognition. It is enough to recognize the identity relationships between proper names in a given text ("the same or not the same") to generate statistics. The adopted stem-based comparisons method seems to be a powerful solution to the problem. Its advantage is that it is able to handle proper names encountered for the first time. Its disadvantage is that it provides good accuracy only for little or medium-sized texts. It is definitely a task for the future to build a dictionary of proper names which can be useful in a broader domain.

How can evaluation methods be improved? The aim of the work was to improve the quality of summarizing in various aspects. However, the existing evaluation methods seemed incapable of measuring issues such as informativity, misinformation and T-grammaticality which are vital for evaluating automatic summaries. The new method was devised to account for these issues in a comprehensive and accountable way. The method combines user feedback concerning a text corpus with methods originating in text linguistics. The need for this is simply a consequence of rethinking

the idea of summary and making the evaluation criteria more "natural" and based on what science already knows about the nature of text.

What does "important information" mean/not mean? Evaluation and tests has also falsified a popular idea that "important information" is very much tided to "facts": numbers, dates, people, places. The research has shown that the story is considered more important than data and that users are often not interested even in the name of the agent. However, in some cases the name of the agent is very important. The conclusion is that only if the name of the person is already known to the user or that person is vital for the story then his name must be included. In other cases it is less obvious. This may also indicate a need for somewhat personalized summarization (which is not a new idea, see Hassel [10]).

What are the limits of extraction technology and how they can be overcome? The method of extraction has its limits. It can be easily deduced from linguistics and seen in the results of people asked to rewrite 1/3 of the original amount of sentences to make a text summary. Sometimes it does not give a correct summary but the score of 80–90% is the highest possible. Research by de Smedt [13] has already shown that the score will be much lower if we reduce the expected summary length below 30%. This means that only abstraction can give a really good ratio of informativity and T-grammaticality to length. The major issues that have to be tackled in order to construct such an algorithm include:

1. Parsing a sentence (finding out the relationships between its parts).
2. Solving surface sentence interdependencies (e.g. pronoun dependencies, the issue of co-reference, relations of time, place, cause, purpose, condition, concession, lexical relations etc.). The complexity and importance of these interrelations is often underestimated by researchers in the field.
3. Summarizing the data by macrooperations (generalization, deletion etc.).
4. Generating a correct text (well-formed on micro- and macrolevel).

References

[1] "summarize" (entry) in Merriam-Webster Online Thesaurus, 15 Jun 2005, http://www.m-w.com/cgi-bin/thesaurus

[2] Van Dijk T. A.: *Some Aspects of Text Grammars. A Study in Theoretical Linguistics and Poetics*, Mouton, The Hague, 1972

[3] Dalianis H., Hassel M., Smedt de K., Liseth A., Lech T.C., Wedekind J.: *Porting and evaluation of automatic summarization*. In Holmboe H. (ed.), Nordisk Sprogteknologi 2003. Arbog for Nordisk Sprakteknologisk, Forskningsprogram 2000–2004, pp. 107–121.

[4] Dalianis H., Hassel M., Wedekind J., Haltrup D., Smedt de K., Lech T. C.: *Automatic text summarization for the Scandinavian languages*. In Holmboe H. (ed.), Nordisk Sprogteknologi, 2002. Arbog for Nordisk Sprakteknologisk Forskningsprogram 2000–2004, pp. 153–163.

[5] Mazdak N.: *FarsiSum – a Persian text summarizer*. Master thesis, Department of Linguistics, Stockholm University, 2004

[6] Pachantouris G.: *GreekSum – A Greek Text Summarizer*. Master Thesis, Department of Computer and Systems Sciences, KTH – Stockholm University 2005

[7] Lin C. Y.: *Training a Selection Function for Extraction*. In the 8th International Conference on Information and Knowledge Management (CIKM 99), Kansa City, Missouri, 1999

[8] Luhn H. P.: *The Automatic Creation of Literature Abstracts*. IBM Journal of Research and Development, 1959, pp. 159–165

[9] Edmundson H. P.: *New Methods in Automatic Extraction*. Journal of the ACM 16(2), 1969, pp. 264–285.

[10] Hassel M.: *Evaluation of automatic text summarization – a practical implementation*. Licentiate thesis Stockholm, NADA-KTH, 2004

[11] Dalianis H.: *SweSum – A Text Summarizer for Swedish*. http://www.dsv.su.se/%7Ehercules/papers/Textsumsummary.html, 2000.

[12] Dalianis H.: *Aggregation in Natural Language Generation*. Journal of Computational Intelligence, Vol. 15, No. 4, 1999, pp. 384–414.

[13] Smedt de K., Liseth A., Hassel M., Dalianis H.: *How short is good? An evaluation of automatic summarization*. In Holmboe, H. (ed.) Nordisk Sprogteknologi 2004, pp. 267-287

[14] Gajecki M.: *Serwer lekskalny jezyka polskiego*. Computer Science, Rocznik AGH, 2001

Appendix. Example of summaries

Na dwa lata pozbawienia wolności w zawieszeniu na pięć lat skazał Sąd Rejonowy w Białymstoku nauczycielkę z Choroszczy oskarżoną o nieumyślne spowodowanie śmierci dwóch dziewczynek. Ponadto orzekł wobec niej zakaz wykonywania zawodu nauczyciela i zajmowania stanowisk związanych ze sprawowaniem opieki nad dziećmi i młodzieżą na trzy lata. Do wypadku doszło 18 maja ubiegłego roku nad Zalewem Siemianówka w Bondarach w woj. podlaskim. Dwie siostry Anna i Małgorzata B. w wieku 11 i 12 lat przebywały w ośrodku na szkolnym biwaku. ***Dziewczynki postanowiły zrobić sobie pamiątkowe zdjęcie nad brzegiem zalewu.*** O zgodę zapytały nauczycielkę. Nad wodę poszły w towarzystwie kilku koleżanek. Pozostała grupa w tym czasie wybierała się do sklepu na zakupy. W pewnej chwili jedna z sióstr poslizgnęła się i wpadła do wody. Druga usiłowała ją ratować. Niestety, obie utonęły. Pozostałe dziewczynki wzywały pomocy. Jeden z wędkujących nieopodal chłopców pobiegł do ośrodka. W pobliżu nie było wtedy nikogo z dorosłych. ***W uzasadnieniu wyroku sędzia podkreśliła, że nauczycielka nie wykonywała należycie swoich podstawowych obowiązków, do jakich należała niewątpliwie opieka nad dziećmi. Wprawdzie tuż przed wyjazdem na biwak dzieci podpisywały regulamin, w którym jeden z punktów kategorycznie zabraniał samodzielnego oddalania się nad wody zalewu, w rzeczywistości jednak takie zdarzenia miały miejsce za zgodą oskarżonej.*** Pozostali opiekunowie trzymali w swoich grupach dyscyplinę, mimo że były to dzieci z klas starszych.

– Jedynie oskarżona pozwalała na tak szeroki zakres swobody swoim podopiecznym – powiedział podczas procesu jeden ze świadków. ***Prokurator, który zażądał dla oskarżonej dwóch lat pozbawienia wolności w zawieszeniu, stwierdził, że wyrok jest zadowalający.*** Obrona zamierza się odwoływać. Wyrok nie jest prawomocny. Piotr Sadziński

Siedem tysięcy km za trzy dolary. ***Grupa kanadyjskich studentów ustanowiła światowy rekord długości przejazdu samochodem napędzanym energią słoneczną. Przez całą podróż auto zużywało 1000 W energii elektrycznej, czyli tyle, ile potrzebuje domowy toster.*** Podróżowali przez Kanadę z przeciętną prędkością około 80 km/godz., pokonując 4376 mil (7416 km). Prawie miesięczna wyprawa rozpoczęła się w Halifaksie i skończyła w Vancouver. ***Studenci z Queen's University w Kingston w Ontario pobili dotychczasowy rekord Australijczyków, który wynosił 2521 mil (4056 km). Samochód o nazwie Radiance czerpie energię słoneczną, która przetwarzana jest na elektryczną służącą bezpośrednio do napędzania silnika.*** Jej nadmiar był przechowywany w baterii akumulatorów, które stanowią rezerwę na pochmurne dni. Przypominający statek kosmiczny czarny samochód ma płaskie nadwozie z prostokątnymi ogniwami słonecznymi. Miejsca w kabinie wystarcza tylko dla jednej osoby, która, ze względów aerodynamicznych, zajmuje pozycję leżącą. Dwaj kierowcy na zmianę przejeżdżali

około 186 mil dziennie (299 km). ***Szef zespołu – James Keirstead podkreślił na zakończenie wyprawy, że na pokonanie trasy z Halifaksu do Toronto auto potrzebowało energii wartości 3 dolarów kanadyjskich.*** Dla kontrastu minivan, którym poruszali się pozostali członkowie zespołu, spalił benzynę o wartości 405 dolarów kanadyjskich. NA PODST. REUTERS OPR. KRU

Oscar za całokształt twórczości. Jak dowiadujemy się nieoficjalnie, Andrzej Wajda został uhonorowany przez Amerykańską Akademię Filmową Oscarem za całokształt twórczości. Składająca się z 39 osób Board of Governors obradowała wczoraj trzy godziny. Było kilku kandydatów do statuetki. Każdy z nich miał w Radzie swojego „referującego". ***Kandydaturę Andrzeja Wajdy uzasadniali podobno znakomity operator John Bailey oraz jeden z najlepszych amerykańskich producentów Saul Zaentz.*** Przypomnijmy też, że polskiego twórcę poparł Steven Spielberg. Jego list podpisało ponad 70 artystów amerykańskich, z Woodym Allenem i Oliverem Stonem na czele. Członków Akademii obowiązuje zachowanie tajemnicy aż do ogłoszenia oficjalnego komunikatu. Jednak z nieoficjalnych informacji dowiadujemy się, że kandydatura polskiego twórcy zyskała w Radzie Gubernatorów bardzo szerokie poparcie. O znakomitej atmosferze wokół naszej kandydatury mówi też dobrze osadzony w hollywoodzkich układach Krzysztof Wojciechowski, który w poprzednich latach przygotowywał oskarowe promocje kilku naszych filmów. Andrzej Wajda jeszcze wczoraj nie wierzył w swoje szanse. – Są znakomici kandydaci do tej nagrody, ludzie, którzy bardzo dużo znaczą w Hollywood – powiedział „Rzeczpospolitej". ***Gdyby nasza informacja potwierdziła się oficjalnie – byłby to olbrzymi sukces polskiego mistrza.*** Jeśli przecieki nie są prawdziwe – pozostaje ciągle wspaniały list Spielberga, będący wielkim hołdem dla naszego mistrza. ***W najbliższą sobotę Andrzej Wajda i producent Lew Rywin wylatują do Rzymu, gdzie na prywatnym pokazie zaprezentują „Pana Tadeusza" papieżowi Janowi Pawłowi II.*** Barbara Hollender