DOI: 10.15439/2015F186
ACSIS, Vol. 5

Automatic Summarization of Polish News Articles by Sentence Selection

Krzysztof Jassem, Łukasz Pawluczuk
Adam Mickiewicz University
in Poznań
ul.Wieniawskiego 1, 61-712 Poznań, Poland
Email: jassem@amu.edu.pl, lp44246@st.amu.edu.pl

***Abstract*—This paper describes the automatic summarization system developed for the Polish language. The system implements sentence-based extractive summarization technique, which consists in determining most important sentences in document due to their computed salience. A structure of the system is presented, as well as the evaluation method and achieved results. The presented attempt is intended to serve as the baseline for future solutions, as it is the first summarization project evaluated against the Polish Summaries Corpus, the standardized corpus of summaries for the Polish language.**

I. Introduction

AUTOMATIC text summarization is a very active research field in recent years. Its purpose is to reduce a text document, by extracting its most important parts in order to create more condensed, but still human-readable form, known as summary. The task consists in the creation of an appropriate computer application and a framework for testing and evaluation.

In this paper we focus on sentence-based extractive summarization using machine learning. We implement well-known techniques, improved and merged into a single summarizing system. The system uses a list of features applied in previous projects, supplemented by new ones, introduced by the paper's authors. Polish Summaries Corpus, a resource created by Ogrodniczuk and Kopeć in [1] has been used as the dataset for training machine learning algorithms. No one has ever used this corpus to create a summarizing system before.

Moreover, an evaluation method has been developed. It is based on the ROUGE summarization evaluation package introduced at Document Understanding Conference (DUC) in 2004, by Chin-Yew Lin [2], who proved it to be a correct measure for the task. We propose to use this evaluation method in future automatic summarization solutions for the Polish language for the sake of objective comparison. The present solution could then serve as the baseline for new systems.

The paper is organized as follows: the rest of the current section describes briefly the aim of the summarization task and main methods in the field. Section 2 provides a review of already existing summarization systems for the Polish language. In section 3 Polish Summaries Corpus is described in detail. Section 4 outlines our solution, it's overall framework, as well as the employed set of features. Section 5 introduces the evaluation methodology and presents our experiments and their results. Eventually, section 6 contains some conclusions and the outline for future work.

A. Aim of summarization

Modern digital technologies, including World Wide Web, result in information excess. Everyday brings vast amount of new on-line information of various type. Processing this continuously growing information databases is not possible by a single human. Automatic summarization is an attempt to confront information processing needs. It is based on the assumption that a computer system can read all data quickly and present its condensed from. Summarization is useful in medicine, law or scientific areas, as well as in everyday life.

Formally, in the area of text summarization, "summary can be defined as a text that is produced from one or more texts, that contains a significant portion of information in the original text(s), and that is no longer than half of the original text(s)" [3].

B. Methods of summarization

Automatic text summarization may be classified according to program's input or output. As regards input, summarization may concern one document or multiple documents (multi-document summarization). Further, in case of multi-document summarization, input data may be *monolingual* or *multilingual*. As regards output, one may distinguish *extracts* and *abstracts*. Mani (2001) claims that "an extract is a summary consisting entirely of material copied from the input" (which in fact can be paragraphs, sentences, phrases, terms or even single nouns) and "abstract is a summary at least some of whose material is not present in the input". Extractive summaries are obviously easier to obtain. Moreover, summaries may be *indicative* or *informative*, which means they can indicate source text's topics and give a brief idea of what the original text is about, or cover the topics in the source text, respectively [3]. Finally, *generic* and *user-focused* (a.k.a. *query-driven*) summaries may be distinguished. *Generic* summaries try to cover all relevant information from the source text, while *user-focused* ones respond to user's information needs expressed as topic or query [3].

Literature often considers automatic summarization a three-stage process. Lloret (2006) names the following steps of the process:

978-83-60810-66-8/$25.00©2015, IEEE

- *interpretation* of the source text in order to obtain a text representation,
- *transformation* of the text representation into a summary representation,
- *generation* of the summary text from the summary representation.

Methods of text summarization may differ as far as the level of processing is concerned: *surface*, *entity*, or *discourse* levels [4]. It is worth noting that there exist systems, which adopt hybrid-approaches.

Surface-level approaches make use of shallow features to analyze information included in a text document. Usually, these features are combined together into a salience function used to extract information. Examples of such features are:

- Thematic features — based on term frequency analysis and statistically salient terms,
- Location features — based on position in text, paragraph or section depth,
- Background features — based on presence of title or headings terms, or a user's query,
- Cue words and phrases — based on presence of special 'bonus' or 'stigma' terms.

Entity-level approaches are based on the internal representation of text. They model text entities and their relationships across a document. Examples of such relationships between entities are:

- Similarity — e.g. vocabulary overlap,
- Proximity — distance between text units,
- Co-occurrence — words occurring in common contexts,
- Thesaural relationship among words — e.g. synonymy, hypernymy,
- Coreference — e.g. anaphora, cataphora, noun phrases,
- Logical relations — e.g. agreement, contradiction, entailment, consistency,
- Syntactic relations — e.g. relations based on parse trees,
- Meaning representation-based relations — e.g. predicate-argument relations.

Discourse-level approaches model the global structure of text, and its relation to communicative goals. Examples of such structures are:

- Format of the document,
- Threads or topics as they are revealed in the text,
- Rhetorical structure of the text.

II. Review of experiments on summarization of Polish Texts

This section covers experiments on automatic summarization for the Polish language, resulting in theoretical works, as well as working implementations. All of them apply extractive methods of summarization.

A. PolSum2 (S. Kulikow)

The first attempt on automatic text summarization for the Polish language was made by Ciura et al. [5] and resulted in the *PolSum* system, which then evolved into *PolSum2*. The system is still available at http://las.aei.polsl.pl/PolSum/. *PolSum2* is an extractive system. It performs various kinds of text analysis (morphological, syntactic, semantic) in order to extract most important sentences from an input document. The system also recognizes anaphora, which results in better coherence between selected sentences.

PolSum2 performs in three stages of summarizing[5]. The first stage, called 'Calling remote analyzer' is intended to call the remote server, which performs text analysis. The *Linguistic Analysis Server* (LAS) is used for this purpose. This tool, created by the same authors, performs linguistic analysis on the levels of: morphological, syntactic and semantic analysis. The syntactic analysis builds a parse tree on the basis of Syntactic Group Grammar for Polish (SGGP) [6]. The system also performs the analysis of anaphoric relations. The seconds stage of summarization process is 'Selecting the essential sentences'. There is no concrete information on the criteria for sentence weighting. The last stage is called 'linearization'. It is designed to create coherent output. Proper forms of words are generated and placed in proper places in sentence. The system also performs homonyms reduction and anaphora substitution for better result reading.

The papers that describe the system do not provide any information about evaluation results.

B. Lakon (A. Dudczak)

Adam Dudczak's *Lakon* is another automatic text summarization system created for the Polish language [7]. It is available on-line at http://www.cs.put.poznan.pl/dweiss/research/lakon/. The system was developed as a result of author's Master Thesis, whose one of main goals was to compare effectiveness of some popular extractive methods for the Polish language. Three methods were developed. They were based on the following heuristics:

- $tf \times idf$ and *Bm25 Okapi* — assumes that words occurrence frequency determines sentence's salience
- sentence's position in text — assumes that most important sentences are often at the beginning of paragraphs,
- lexical chain — assumes that relations across sentences determine their salience.

The system was evaluated on the corpus created from 10 manually summarized newspaper articles. 60 volunteers manually created totally 285 summaries of these articles. Evaluation results indicated that the most effective features were words occurrence frequency and sentence's position. The lexical chains method was proved to be worse than the others.

C. Summarizer (J. Świetlicka)

Świetlicka's Summarizer [8] is the latest tool created for Polish. It is available on-line at http://clip.ipipan.waw.pl/Summarizer. This solution is the most similar to the one proposed here. It uses various machine learning methods for training an extractive summarizer based on a set of sentence's features. These features include:

- LLR — *Log Likelihood Ratio*,
- $tf \times idf$,

- Sentence's centrality,
- Occurrence of characteristics phrases — bonus and stigma words, popularity of one or two first words of a sentence,
- Similarity to the title — indicating occurrence of words from the title in a sentence,
- Number of words starting with uppercase — indicating Named Entities,
- Number of tokens that are not proper words — i.e. punctuation or numbers,
- Localization — position of the sentence in paragraphs and position of the sentence in the whole text,
- Length of sentence,
- Length of paragraph,
- Length of text,
- Type of sentence — based on the last token: declarative, interrogative, imperative.

A number of tests were performed on different subsets of these features. The author used about 13 different machine learning algorithms in order to compare their effectiveness. The corpus was created by the author on his own and contains 102 newspaper articles for training and 67 articles for evaluation.

Świetlicka's Summarizer also performs simple summary linearization. It consists of three steps. At first, sentences are sorted in the order of their appearance in the document. Secondly, fragments in parentheses are removed in order to make sentences shorter. Lastly, some special words are removed from the beginnings of sentences, such as: therefore, moreover or however.

The discussed work contained the following conclusions:

- localization-based features, particularly sentence position in the paragraph and the whole document, tend to be the most important ones,
- sentence centrality feature is also very effective,
- cue words feature are not so effective,
- machine learning algorithms tend to be an effective solution for automatic summarization. Using a set of features result in better quality than using each separate feature.

III. Polish Summaries Corpus

Polish Summaries Corpus is a resource created by Ogrodniczuk and Kopeć in 2014 [1]. Its aim is to provide a high quality corpus containing manual summarization examples. The corpus forms a significant facilitation for further researchers, who can build their own summarization tools based on this corpus, as well as evaluate them. Ogrodniczuk and Kopeć notice that previous works on automatic summarization in the Polish language lacked a common corpus and a common evaluation method, therefore their results are not comparable. *Rzeczpospolita corpus* — a collection of articles from the Web archive of a Polish newspaper [9] was used as the base corpus for Polish Summaries Corpus.

Polish Summaries Corpus contains 569 text documents divided into 7 categories: Society and Politics, Sport, Economy, Culture news, Law, National news and Science and Technology. All these texts have been manually summarized by independent annotators. All 569 documents have the extractive summaries and 154 have also the abstractive summaries. For each document in the corpus 5 independent propositions of summarization have been created. Each proposition of summarization contains 3 summaries of a given text of the approximate length of 5%, 10%, 15% of the original, respectively. The summaries are included in one another: 10% summary contains only fragments from previously selected 20% summary and so on. Therefore, the corpus size is 8355 summaries.

IV. The proposed solution

The solution presented here implements sentence-based extractive summarization. It consists of two main components: linguistic analysis and summarization application. The latter component selects essential sentences and generates the result summary. The summarization component appplies neural networks as a machine learning algorithm. The Open Source implementation — PyBrain[10] was used.

A. Methodology

The linguistic analysis component performs various kinds of text analysis. The input document is divided into paragraphs, sentences and tokens. Subsequently, lemmatization is performed, parts of speech, named entities and headers are determined. Finally, the internal document model is created and transferred to the summarization component.

The summarization component works as a three-stage process. The first stage computes feature values for each sentence in the document. The second stage is sentence weighting based on the previously trained machine learning model and computed features. In the third stage, the summary is prepared according to the obtained sentences weights. This stage includes the sorting of result sentences, according to their order in the original document.

B. Description of features

This section describes each feature used in the system. Selection of the features was based on literature [3], [4], [11], [7], [8] as well as a few new ideas. The complete list of used features includes:

- *TfIdf* — sum of *term frequency – inverse document frequency* value for every word in sentence,
- *Centrality* — arithmetic average of sentences similarity to every other sentence in the document. Cosine similarity is used as a similarity measure between two sentences,
- *SentLocPara* — position of a sentence in the paragraph: in the first, second or third of equal parts,
- *ParaLocSection* — position of the paragraph in the document: in the first, second or third of equal parts,
- *SentSpecialSection* — occurrence in a special section like the beginning (introduction) or ending (conclusion) of document,

$$\text{ROUGE-N} = \frac{\displaystyle\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n\in S} Count_{match}(gram_n)}{\displaystyle\sum_{S\in\{ReferenceSummaries\}} \sum_{gram_n\in S} Count(gram_n)} \quad (1)$$

- *SentInHighestTitle* — number of words from heading or title in the sentence,
- *ParaLength* — paragraph length: short (up to 1 sentence), average (2–5 sentences) or long (more than 5 sentences),
- *SentLength* — sentence length: short (up to 7 words), average (7–14 words) or long (more than 14 words),
- *SentType* — type of the sentences, based on its last punctuation mark: declarative, interrogative or imperative.
- *MetaInfo* — sentences not referring to the document content, i.e.: an information about document's author or photo signatures,
- *AvWordLength* — the average of words lengths in sentences,
- *Verb* — existence of the final verb,
- *Nouns* — number of nouns in sentence,
- *Pronouns* — number of pronouns in sentence,
- *SentInHighestPname* — number of Named Entities in the sentence as found by a naive method, recognizing Named Entity as a word starting with capital letter,
- *NER* — number of Named Entities in the sentence as found by NERf Named Entities Recognition tool [12],
- *NERTf* — sum of every Named Entity frequency in the whole document, occurring in given sentence,
- *PersNameNE* — number of recognized NE of the "person" type,
- *OrgNameNE* — number of recognized NE of the "organization" type,
- *PlaceNameNE* — number of recognized NE of the "place" type,
- *DateNE* — number of recognized NE of the "date" type,
- *GeogNameNE* — number of recognized NE of the "geography" type,
- *TimeNE* — number of recognized NE of the "time" type.

The features applied by authors of this paper, which were not mentioned in the referred works, are: *MetaInfo*, *AvWordLength*, *Verb*, *Nouns*, *Pronouns*, *NER*, *NERTf*, *PersNameNE*, *OrgNameNE*, *PlaceNameNE*, *DateNE*, *GeogNameNE* and *TimeNE*.

V. Evaluation

A. Evaluation method ROUGE (DUC conference)

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation [2]. It was introduced by Chin-Yew Lin at Document Understanding Conference (DUC) in 2004 and since then it has became the standard method for the evaluation of automatic summarization systems. It provides a set of measures to automatically determine the quality of summary in comparison to ideal summaries created by humans. The measures are based on overlapping units such as n-grams, word sequences and word pairs. ROUGE has been proved to be highly correlated with human judgements. This section describes ROUGE-N methods, which were proved to work well in single document summarization tasks.

ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries [2].

It is computed using the (1) formula, where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. It is worth noting that the denominator of (1) increases if more than one reference documents are used. Moreover, larger weight is assigned to matching n-grams occurring in multiple references, so if words are shared by more references, ROUGE-N favors them.

B. Experiments and Results

A number of experiments were performed. Different subsets of features were used in order to achieve the best results in summarization. Every learned model for each features susbset was evaluated with ROUGE-1, ROUGE-2 and ROUGE-3 methods. Random summarization was used as a baseline. Features were divided into subsets, as follows:

- *Sub1* $\in \{TfIdf, ParaLength, Centrality, SentType, SentSpecialSection, SentInHighestTitle, SentLength, SentLocPara, ParaLocSection\}$
- *Sub2* $\in$ *Sub1* $\cup \{Pronouns, MetaInfo, Verb, Nouns, AvWordLength\}$
- *Ner1* $\in \{SentInHighestPname\}$
- *Ner2* $\in \{NER, NERTf\}$
- *Ner3* $\in \{OrgNameNe, GeogNameNe, DateNe, PlaceNameNe, PersNameNe, TimeNe\}$

Evaluation results are placed in Table I. It is clear that almost every subset of features used in experiments gave nearly the same results, which were about 15% better than the baseline, according to the F-1 score. No feature subset performed clearly better than the others. New features, included in *Sub2* raised the score slightly, just as dividing the Named Entities information into categories did (*Ner3*). In fact, using the NER tool, instead of naive methods tends to give slightly better results in summarization. Summing up, the best results in ROUGE-1, ROUGE-2 and ROUGE-3 were achieved using the largest subset of features. The experiments have shown that developing new features may be quite useful, but there is no single feature that separately raises the score significantly.

VI. Conclusion and Future Work

In this article, we have presented the document summarizing approach for the Polish language. It is based on sentence extraction and applies neural networks as a machine learning

TABLE I
EXPERIMENTS' RESULTS.

	ROUGE-1			**ROUGE-2**			**ROUGE-3**		
	Recall	Precision	F-1	Recall	Precision	F-1	Recall	Precision	F-1
RANDOM	0.29	0.34	0.30	0.14	0.17	0.15	0.13	0.15	0.13
$Sub1$	0.45	**0.45**	0.44	0.34	**0.33**	0.33	0.33	**0.32**	0.32
$Sub2$	0.47	**0.45**	0.45	0.35	**0.33**	0.33	0.35	**0.32**	0.33
$Sub1 \cup Ner1$	0.44	0.42	0.43	0.32	0.29	0.30	0.31	0.28	0.29
$Sub1 \cup Ner2$	0.49	0.43	0.45	0.37	0.31	0.34	0.37	0.31	0.33
$Sub1 \cup Ner3$	0.49	0.42	0.45	0.37	0.30	0.33	0.36	0.29	0.32
$Sub2 \cup Ner1$	0.48	0.42	**0.46**	0.37	0.32	0.34	0.36	0.31	0.33
$Sub2 \cup Ner2$	0.47	0.42	0.44	0.35	0.3	0.32	0.34	0.29	0.31
$Sub2 \cup Ner3$	**0.51**	0.43	**0.46**	**0.39**	0.32	**0.35**	**0.39**	0.31	**0.34**

algorithm. This approach seems to be promising in achieving acceptable summarizing method for the Polish language, however there are some difficulties in choosing the proper features set and tuning machine learning algorithm. We conclude that there is still much work to do in the field. We hope that our approach will serve as a inspiration, as well as a baseline for the future research at the task of automatic summarization for Polish.

REFERENCES

[1] M. Ogrodniczuk and M. Kopeć, "The polish summaries corpus," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, N. C. C. Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Reykjavik, Iceland: European Language Resources Association (ELRA), may 2014. ISBN 978-2-9517408-8-4

[2] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. ACL workshop on Text Summarization Branches Out*, 2004, p. 10.

[3] E. Lloret, "Text summarization: An overview," [on-line] http://www.dlsi.ua.es/~elloret/publications/TextSummarization.pdf.

[4] I. Mani and M. Maybury, *Advances in Automatic Text Summarization*. MIT Press, 1999. ISBN 9780262133593

[5] M. Ciura, D. Grund, S. Kulików, and N. Suszczanska, "A system to adapt techniques of text summarizing to polish," in *International Conference on Computational Intelligence, ICCI 2004, December 17-19, 2004, Istanbul, Turkey, Proceedings*, A. Okatan, Ed. International Computational Intelligence Society, 2004. ISBN 975-98458-1-4 pp. 117–120.

[6] N. Suszczańska and M. Lubiński, "Polmorph, polish language morphological analysis tool," in *19th IASTED International Conference APPLIED INFORMATICS - AI'2001, Innsbruck (Austria)*, 2001, pp. 84–89.

[7] A. Dudczak, J. Stefanowski, and D. Weiss, "Automatyczna selekcja zdań dla tekstów prasowych w języku polskim," Institute of Computing Science, Poznan University of Technology, Poland, Technical Report RA-03/08, 2008.

[8] J. Świetlicka, "Metody maszynowego uczenia w automatycznym streszczaniu tekstów," Master's thesis, University of Warsaw, 2010.

[9] D. Weiss, "Korpus rzeczpospolitej," [on-line] http://www.cs.put.poznan.pl/dweiss/rzeczpospolita.

[10] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," *Journal of Machine Learning Research*, 2010.

[11] I. Mani, *Automatic Summarization*, ser. Natural language processing. J. Benjamins Publishing Company, 2001. ISBN 9789027249869

[12] J. Waszczuk, K. Głowińska, A. Savary, and A. Przepiórkowski, "Tools and methodologies for annotating syntax and named entities in the National Corpus of Polish," in *Proceedings of the International Multiconference on Computer Science and Information Technology (IMCSIT 2010): Computational Linguistics – Applications (CLA'10)*. Wisła, Poland: PTI, 2010, pp. 531–539.