Coreference-based Content Selection for Automatic Summarization of Polish News

Mateusz Kopeć

Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland

Abstract. This article addresses the problem of automatic summarization of press articles in Polish. Main novelty of this research lays in the proposal of a summarization algorithm relying mainly on coreference information. Such system was never implemented for Polish language. Moreover, the article contains first evaluation of several Polish summarizers on the same dataset.

First, three publicly available summarization systems for Polish are presented. Second, the new coreference-based extractive summarization system EMILY is introduced. EMILY's algorithm utilises advanced third-party preprocessing tools to extract coreference information from the text to be summarized. This information is transformed into a complex set of features related to coreference concepts (mentions and coreference clusters) that are used for training (on the basis of a manually prepared gold summaries corpus) the machine learning summarization system.

Newly proposed solution is compared to the three previously described summarization systems and a BASELINE system, which produces a summary by extracting the beginning of the original text. A method is proposed to normalise the length of summaries for the three reference systems to make the comparison fair. The evaluation is performed on the POLISH SUMMARIES CORPUS, a large corpus of middle-length press articles.

The conclusion is that the BASELINE summarization system obtains best scores, especially for the summaries of a very limited length. EMILY's performance is mediocre, yet it outperforms one of the competing systems in a statistically significant way.

1 Introduction

According to the Oxford Advanced Learner's Dictionary[1] *a summary* is *a short statement that gives only the main points of something, not the details.* Nowadays, the amount of information available is astonishingly big, mainly due to

[1] See `http://www.oxforddictionaries.com/definition/learner/summary`.

its electronic form of storage. This form allows automatic computer processing of that information, making it feasible for humans to take advantage of the so-called *big data* and benefit from the knowledge accumulated by automated systems and presented to the reader in a concise way.

Narrowing the focus to the natural language data, we still observe an overwhelming amount of texts available in the Internet. It is impossible for anybody to monitor and read all the information appearing at information portals, even when only selected subjects are of one's interest. Available information is often too broad.

One of the solutions to such problems is *automatic summarization*, performed on a large volume of documents, which allows for controlled information selection and compression. A system which performs the automatic summarization should be a replacement for an expert. Its role is to process many documents on the selected topic and present a summary, describing most interesting facts.

In this article I focus on the possibility of using coreference information as the main factor influencing the automatic summarization process. Moreover, I consider only single document summarization of press articles, written in Polish language. Such approach was not yet investigated by any researcher. The main reason behind the language choice (despite the fact it is my native one) is recent appearance of the required toolset for Polish: the first tools performing coreference resolution [1, 2, 3, 4, 5] as well as the corpus of manually constructed summaries [6], which may be used both for development and evaluation of new summarization methods.

1.1 Coreference-related definitions

Coreference is a linguistic relation, occurring between two or more expressions in a text referring to the same object. For example in the text:

Bill came. He was upset.

both *Bill* and *He* refer to the same entity. All expressions referring to the same object in a text are connected with coreference relation, therefore they form a *coreference cluster*, sometimes also called a *coreference chain* (a sequence of cluster elements in the order they appear in the text). The expressions which are subjects of the coreference relation are called *mentions*. In this article we will consider only noun-phrase mentions, including pronouns and zero (or null) subjects. The phenomenon of the zero subjects in Polish is very important to take into account, as about 30% of sentences use such subjects to express the central entity of the sentence [7].

For the scope of this article we will focus on single-document coreference, although multi-document relations are also studied by many researchers.

2 Related Work

First work on the automatic summarization appeared in the late '50s published by Luhn [8]. The early research involved performing simple statistical experiments on extraction summarization. A summary was created by selecting the most important sentences from the original document, such extraction of full sentences was a way to avoid the problem of generation of syntactically and semantically correct sentences.

Next years of the summarization research (e.g. [9]) involved improvements in sentence weighting techniques by introducing more weighting factors, like position of the sentence in text, appearance of words from the title of the document or presence of predefined cue-words. It is worth to notice that in these times the balance in weighting different features was maintained manually by researchers, who selected weights via experiments. These approaches were gradually replaced by supervised machine learning techniques to automatically deduce the optimal weights of various factors of sentence importance. For that purpose, corpora of human-created summaries were produced, for example Ziff-Davis corpus by Marcu [10].

Later on, sentence extraction was conducted via new methods, for example by clustering sentences and then selecting a representative sentence from each cluster, to limit redundancy in the final summary (e.g. [11]). Methods using pure lexical frequency statistics were enriched by knowledge sources, such as thesauri, allowing for detection of the same entity mentioned using different words throughout the document (e.g. [12]). Lexical or coreference chains were utilised to signal the main text topic and its side-topics.

In this section I am going to describe the automatic summarization approaches which build on coreference information to extract most important content from the source text. As no similar experiments were made up to date for Polish language, I am also going to provide a background study of several summarization attempts for Polish, as they will be used for an evaluation of the newly proposed algorithm.

2.1 Summarization using coreference

The underlying assumption for using coreference for summarization is that coreference chains allow to identify the most important entities appearing in a given document, which often cannot be done using purely lexical data (as entities are represented most often using several different phrases, pronouns or zero subjects). Moreover, efficient coreference resolution may be used to produce entity-focused summaries, for example by extracting only these sentences from the source document, which contain a mention coreferent with given entity.

Most articles about using coreference information for summarization can be divided into two types:

1. coreference-based summary content selection,
2. coreference-based extractive summary revision.

The focus of this article is the first type, therefore only such research is presented in this section.

The first work which connects summarization with coreference resolution was published in 1998. Baldwin and Morton [13] study query-sensitive summarization by sentence extraction. The target summary is supposed to be indicative and show whether the summarized document is relevant to user's query. Identity and part-whole relations are identified between mentions in the document and the query. Mentions are defined as proper names (compared using dictionaries, acronym tables, character sequence similarity), verbs (compared by their co-occurrence frequency or presence of the same arguments), adjectives and nouns (compared by lemma and part of speech equality). On the basis of detected coreference chains, sentences are selected to the summary in a greedy fashion, to cover as many coreference chains containing mentions from the query as possible.

Summarization system by Azzam et al. [14] creates generic single-document summary of a text by selecting one of the automatically detected coreference chains in it. The 'best' chain is chosen heuristically, and the final output of the system is the concatenation of a subset of original sentences containing elements from the chain.

Stuckardt in [15] analyses an application of coreference to the summarization task. The author suggests that both use cases benefit the most from connecting lexically informative mentions with non-informative ones with coreference relations, e.g. it is more useful to know which proper noun is referred to by a pronoun, than that two pronouns co-refer.

The articles [16, 17] focus on the task of producing 10-word single-document summaries. Their system outputs most important noun phrases from the original text, selecting them on the basis of coreference chains they belong to. The chains are ranked according to their size and additional bonus for having an element in the first two sentences of the text. Then, the longest mention from each chain is added to the summary in the order of chain ranks, until desired summary size is obtained.

Steinberger et al. [18] extend a summarization method based on Latent Semantic Analysis (LSA) with automatically obtained coreference information. Novel approach of Steinberger et al. is to add coreference information to the word-sentence matrix in one of two ways: either replace each mention with the first mention in its coreference chain or add the chain identifier as a new feature to the matrix. Several experiments show, that only the second approach improves the performance of the original algorithm.

Hendrickx et al. [19] present an addition of coreference information and sentence compression to a previously introduced graph-based query-focused multi-document update summarization system. Their original system represents sentences as nodes in several graphs, which either model relevancy or redundancy of information in two sentences connected by an edge. They use two relevance graphs: for query-relevance (cosine similarity to query) and relatedness (cosine similarity of sentences from the same source document) and two redundancy graphs: cross-document relatedness (cosine similarity between sentences coming

from different documents) and redundancy (cosine similarity to known information document set). Such graphs are then used to choose most salient sentences, combining information from different types of edges using weights tuned via genetic algorithm. The coreference addition introduces new relevance graph to the system, which simply connects sentences which contain coreferent mentions (with edge weigh derived from the number of such mentions). The scope of coreference resolution concerns only noun phrases.

Work described in [20] concerns a task different from the others presented in this section. The task is to summarize research papers by extracting contexts in which they are cited. There is no final summary produced, the final output of their algorithm is a set of citation contexts. Such task may not be directly compared to news summarization, yet in documents published in the web, hyperlinks may play similar role to citations in research papers.

Smith et al. [21] propose an extraction-based single document summarizer COHSUM, which selects content using coreference information only. Their key idea is to treat sentences as nodes in a graph, connected by coreference relations between mentions contained in these sentences. Nodes in this graph are then scored with the help of an algorithm similar to TextRank [22], which is a variant of famous PageRank algorithm [23]. Additionally, a representative mention (most elaborate) is heuristically chosen for each coreference chain. Only the sentence with such representative mention has both outgoing and incoming links, otherwise only incoming links are introduced. Using such graph, PageRank scores nodes for importance, and COHSUM extracts most highly ranked sentences to the summary in a greedy fashion.

The impact of using coreference resolution to help NLP applications is studied by Mitkov [24]. One of the applications described in the paper is the automatic summarization: the authors test whether a keyword-based summarizer [25] benefits from having the coreference information (produced automatically by BART system [26]).

2.2 Summarization of Polish texts

Not much research was done up-to-date in the area of automatic summarization for Polish texts, especially there were no attempts to benefit from coreference information for summarization. However, several summarization tools for Polish which are publicly available do exist and will serve as the reference point in the evaluation of newly proposed method. These are presented in more detail in next sections.

Dudczak Adam Dudczak researched automatic summarization of Polish news in his master's thesis [27], as well as in several other publications: [28, 29, 30]. His summarization process begins with heuristic paragraph, sentence and word segmentation, as well paragraph title detection. Then one of six methods of sentence selection is used:

- Selecting the beginning of the original text, up to the desired size of the summary.
- Random choice of sentences.
- Relying on sentence position in a paragraph and paragraph position in text – sentences are ordered based on these two properties: most important is the first sentence of the first paragraph, then first sentence of the second paragraph, etc., and after first sentence of the last paragraph comes second sentence of the first paragraph and so on; author reports best scores were obtained by this sentence selection method.
- Using TF-IDF or OKAPI BM25 word scores to rank sentence importance: sentence score is a sum of scores of its lemmatized words (lemmatization via Morfeusz by [31]); reference corpus for two metrics is the Polish Wikipedia. Some parameters are available: whether to score all words or only nouns, whether to score only several words of highest score in each sentence or whether to reject words below a specified threshold from the sum.
- Utilising lexical chains information: the algorithm uses a thesaurus (with synonymy, hyponymy and hypernymy relations) from `synonimy.ux.pl` webpage[2]. Lexical chains of words related in the thesaurus are created and sorted by quality, the longer and more homogeneous first. Best chains are used to score sentences their elements belong to.

Świetlicka Joanna Świetlicka implemented sentence extraction summarization system based on machine learning in her master's thesis [32]. Her implementation uses TAKIPI [33] for segmentation and tagging. Sentences are selected to summary on the basis of a machine-learned combination of the following features:

- LLR – average log-likelihood ratio of words in the sentence. LLR values are calculated using 30 million word subcorpus of IPIPAN Corpus [34].
- TF-IDF – average TF-IDF score of words in the sentence, calculated with the use of the same corpus as the previous feature.
- Sentence centrality – average similarity of the sentence compared to each other sentence. Similarity is calculated as cosine between sentence TF-IDF vectors. There is also another version of this feature, where sentence are clustered according to the defined similarity and centrality score is only measured comparing the sentence to other sentences in its cluster.
- Appearance of words indicating importance (based on a manually created word list), such as „ważny" ('important') or „pierwszy" ('first').
- Appearance of words indicating non-importance (based on a manually created word list), such as „zresztą" ('besides') or „także" ('also').
- Machine-learned score of sentence importance calculated on the basis of its first word or two first words.
- Presence of a word from the title in the sentence.
- Similarity of the sentence and the title (as in centrality).

[2] The site unfortunately does not exist now, however the last versions of its resources are available at `https://dobryslownik.pl/pobierz/`.

- Proportion of capitalized words in the sentence.
- Proportion of non-word tokens in the sentence (e.g. punctuation or numbers).
- Position of the sentence in text and in paragraph.
- Length of the sentence, paragraph, text.
- Last word of the sentence.

Words from a manually created stop list (201 entries) are ignored by most features, there is also a possibility to ignore all words except nouns, as in Lakon.

Sentences are ranked by the learned model on basis of presented features and these with highest score are selected to the summary. However, some smoothing is applied prior to the selection, as sentences adjacent to highly-scored ones obtain some small bonus to their score. Finally, sentence compression is performed: fragments in parentheses are removed, as well as some rhetorical constructions at the beginning of a sentences (e.g. „zatem" ['therefore']).

OpenTextSummarizer OpenTextSummarizer is a multi-language open source tool for summarizing texts, created by Rotem [35]. The tool is used as a benchmark in several publications and includes a setting for Polish language, therefore it is worth to be compared with other, Polish-specific summarization methods.

The algorithm of OTS is very simple. It starts with sentence segmentation, based on a heuristic of chosen sentence-ending characters with a list of exceptions, i.e. specific phrases which contain sentence-ending characters, but do not indicate sentence boundaries. Then the text is stemmed and stop-words are removed. All tokens are assigned a score equal to their count in the whole document, and each sentence is assigned weight equal to sum of scores of all its tokens. Summary is composed from the sentences with the top weights. Size of summary is configurable by the user as the percent of total sentence count of the source document.

Support for Polish language is rather limited – the only customisation is 29-word stop-list. In comparison, English settings also include stemming rules, synonyms and past-forms lemmatization for a number of verbs.

3 Problem Statement

The problem stated in this article is to create a summarization tool, capable of creating summaries of a given ratio, specified as the percentage of the word count of the original document. The summarization task is defined as taking a raw text input and producing a raw text output. The evaluation of a summary should be based on comparing these raw texts. It shouldn't rely on the way gold or system summary was created, as it narrows the possibility of comparing systems, which produce summaries using different means. For example, both Dudczak and Świetlicka created sentence-extraction based systems and evaluated them by comparing with sentence-extraction based gold summaries corpora. This

prevents from comparing their results with other systems, not using sentence extraction.

The target size of the summary is set in the number of words, again to allow for fair comparisons. Using for example the number of sentences would possibly favour the systems producing longer sentences and therefore possible to convey more information (see Section 6.2 for information, how that target was reached for existing systems).

4 Dataset

The dataset used in this article is the Polish Summaries Corpus, a resource created recently to allow for thorough evaluation of single-document automatic summarization systems for Polish news articles. The corpus is available to download at `http://zil.ipipan.waw.pl/PolishSummariesCorpus`, the version used in this article is `1.0`.

The corpus contains manual single-document summaries of press articles coming from the Rzeczpospolita Corpus (RC) by Weiss [36] — a collection of articles from the Web archive of Rzeczpospolita, a nationwide Polish daily newspaper. PSC contains 569 original texts, having from 1000 to 4000 words and coming from 7 sections of the newspaper (see Table 1). On average, a text contains approximately 100 sentences. All 569 text have extractive summaries, 154 out of 569 also have abstractive summaries.

Text domain	**Abstractive corpus**	**Extractive corpus**
Social and political	22	393
Sport	22	36
Economy	22	34
Cultural news	22	32
Law	22	26
National news	22	24
Science and technology	22	24
Total	**154**	**569**

Table 1. Selected domains

Manual summarization was conducted by 11 annotators, who were randomly assigned texts for summarizing. 5 independently created versions of each manual summary were created following the research by Nenkova, where 4 to 5 summaries were said to provide an optimal balance of annotation effort and reliability for the Pyramid method evaluation (see for example [37]).

4.1 Extractive summaries

Annotators were instructed to create three extractive summaries of a given text, each constituting approximately 20%, 10% and 5% of the word count of the original. Minor (a few word-length) deviations were acceptable. Only original words and punctuation in the original order had to be used (so that annotators could e.g. select just the superordinate clause and a finishing dot, removing the less important part of a sentence such as subordinate clauses, interjections, excessing adjectives etc.) As phrases could be selected and sentences combined, lower case start of the sentence or an upper case character in the middle of the resulting sentence was acceptable. The sequence of summaries was forced to be inclusive, i.e. the 10-percent summary had to use only fragments previously selected for a 20-percent summary — and, similarly, the 5-percent summary had to use only fragments previously selected for a 10-percent summary. In this way a partial importance ranking of text spans could be inferred.

4.2 Abstractive summaries

Similarly to the previous task, annotators were instructed to create 3 abstractive summaries of a given text, each constituting approximately 20%, 10% and 5% of the word count of the original, with acceptable minor deviations in word count. Contrary to extractive summaries, abstractive summaries did not have to contain fragments of original texts and could express the same ideas "in own words" of an annotator. Differently from the extractive, longer summaries did not have to contain fragments of shorter ones, but they could.

4.3 Development and test set

Polish Summaries Corpus contains 70 interviews, which are difficult to process for non-specialized coreference resolvers, therefore were rejected from this study. To allow for evaluation in comparison to abstractive, well formed summaries, as justified in the Section 3, all the texts with abstractive summaries were left for final evaluation. Therefore test set contained 154 texts from 7 domains (see Table 1).

Development and training set was composed from all other texts in the PSC, discarding interviews, which resulted in 345 texts total.

5 Solution

The summarization tool is called Emily and was written in Java, its source code is publicly available at Git repository: `http://git.nlp.ipipan.waw.pl/summarization/emily`. The exact version which may be used to replicate the experiment has `1.0` tag. Details of the algorithm behind the tool are described in this section.

5.1 Algorithm

The algorithm relies on dividing the content of the original text into units. In the current implementation, these are either clauses (Emily-C) or full sentences (Emily-S), defined in the tool settings. Each unit is assigned a score based on machine learned model, then units with the highest scores are concatenated to create final summary, until desired summary size is obtained.

The model which scores each unit from the original text is a regression model. Unit examples from the training corpus have their scores calculated based on manual summaries they belong to; each unit is connected to a set of its features. The model learns how to calculate unit scores for new, unseen examples, relying only on their features. This approach is similar to Świetlicka's, yet she used only 1 or 0 as the unit (in her case: a sentence) score, as she used a training corpus with single summary for each training document. I decided to use different unit scoring procedure.

Each document in the training corpus has 15 extractive summaries: 5 independent summaries for each ratio: 5%, 10% and 20%. As the manual extractive summaries were annotated without any constraints, every single character of the original document may either by part of one of the summaries or not. As there are 15 summaries, every character may be in any number of summaries from 0 to 15. The score of each unit is calculated based on its characters: a unit gets one point for each its character, which was included by a single annotator in a single summary. It means, that the maximal number of points a unit may receive is n*15, where n is the character count of the unit.

After each unit in a single text gets its points calculated according the procedure presented above, the unit scores are standardized, i.e. scaled to have mean of 0 and a standard deviation of 1. After such standardization applied in every training text each training example (i.e. a unit) has its gold score attached.

5.2 Preprocessing and unit extraction

The assumption of implemented system is advanced NLP preprocessing, producing:

- sentence, token and paragraph segmentation (Concraft [38] tagger was used for experiments described in this article),
- morphosyntactic analysis and tagging (provided by Morfeusz [31], Maca [39] and Concraft),
- shallow parsing (implemented in Spejd [40])
- named entity recognition (produced by Nerf [41]),
- mention detection (given by Mention Detector [1]),
- coreference resolution (performed by Bartek [1]).

To produce these annotation layers, I used Multiservice [42], a platform maintained by the Institute of Computer Science Polish Academy of Sciences. This platform provides online access to various NLP tools, which may be used both manually and by machine-friendly API.

If the clause unit approach is used, the clause splitting algorithm was needed. It was implemented as in MENTION DETECTOR: First, the sentence was split into candidate clauses at any of the following tokens: *i* (and), *albo* (or), *lub* (or) and characters: comma, semicolon, colon, parentheses, hyphen, en-dash, quote. Then, scanning from the beginning of the sentence, each clause was checked for the presence of a verb. If none were found, it was merged with the next clause. Moreover, clause splits were not allowed inside a syntactic word or group.

For performance reasons during experiments EMILY performed NLP preprocessing once for all the text and then used cached data, however in standard usage input text is sent to MULTISERVICE for preprocessing, which requires internet connection.

5.3 Features of a unit

Each unit was described using a set of mention- and coreference-related features defined in this section. To present these features, it is necessary to define several variables:

- normalization type N_T: words, characters,
- set characteristic type S_T: minimal, maximal, average,
- mention type M_T: any, zero subject, pronoun, singleton,
- non-singleton mention type MNS_T: any, zero subject, pronoun,
- cluster type C_T: any, singleton.

With such definitions, we are able to present the features in a more concise way. Mention-related features of a unit are:

1. number of mentions of type M_T which are inside given unit (4 features),
2. number of mentions of type M_T which are inside given unit, divided by the N_T size of the unit (4*2=8 features),
3. boolean feature, telling whether unit contains at least one mention of type M_T (4 features),
4. the S_T number of N_T in mentions of type M_T inside the unit (3*2*4=24 features),
5. the S_T number of N_T in mentions of type M_T inside the unit, divided by the N_T size of the unit (3*2*4*2=48 features).

Based on mentions inside the unit, coreference chains to which these mentions belong may be obtained. For such coreference clusters the following features were defined:

5. number of clusters of type C_T with at least one mention inside the unit (2 features),
6. number of clusters of type C_T with at least one mention inside the unit, divided by the N_T size of the unit (2*2=4 features),
7. boolean feature, telling whether unit contains at least one mention of a cluster of type C_T (2 features),

8. the S_T number of mentions in clusters of type C_T with at least one mention inside the unit (3*2=6 features),
9. the S_T number of mentions in clusters of type C_T with at least one mention inside the unit, divided by the N_T size of the unit (3*2*2=12 features),
10. the total number of mentions of type M_T inside clusters of type C_T with at least one mention inside the unit (4*2=8 features),
11. the total number of mentions of type M_T inside clusters of type C_T with at least one mention inside the unit, divided by the N_T size of the unit (4*2*2=16 features).

The last set of features concerns only the non-singleton coreference clusters intersecting given unit:

12. for coreference clusters of each mention of type MNS_T: the number of mentions being first in their coreference chains (3 features),
13. for coreference clusters of each mention of type MNS_T: the number of mentions being first in their coreference chains, divided by the N_T size of the unit (3*2=6 features),
14. for coreference clusters of each mention of type MNS_T: the S_T position of a mention from the unit in its coreference chain (3*3=9 features),
15. for coreference clusters of each mention of type MNS_T: the S_T position of a mention from the unit in its coreference chain, divided by the N_T size of the unit (3*3*2=18 features).

In total there are 174 coreference-related features. In addition to these, several standard features were implemented:

16. position of unit in text – from 0 to the total number of units,
17. position of unit in its paragraph,
18. number of N_T in unit (2 features),
19. number of N_T in the paragraph containing unit (2 features),
20. number of N_T in the sentence containing unit (2 features),
21. boolean feature, telling whether unit's last character is a letter,
22. boolean feature, telling whether unit's first character is a letter,
23. boolean feature, telling whether unit is all capital letters.

The main reason to include those 11 features was a conclusion from initial experiments, based on two observations:

- It is well known, that the baseline of taking the first part of a press article as its summary is very hard to outperform. This was confirmed during the experiments. Therefore several simple features about paragraph/document position were introduced.
- Texts in PSC often contain paragraph titles, author or photographer names and similar rather short spans of text. These were very rarely selected to the summary by human annotators, at the same time it would be difficult for the system to identify such units without the knowledge about their last character (most of them ended with a letter, in contrary to standard sentences, ending with punctuation signs), word/character length (they were mostly short) and capitalization (author names were mostly fully capitalized).

5.4 Performance on training set

Two settings were tested:

- with sentence units (34676 in the training data),
- with clause units (48331 in the training data).

In each setting a dataset was extracted from the training corpus. Two machine learning algorithms were applied to that data:

- linear regression,
- regression tree.

Implementations come from WEKA [43] with the default parameter values. The higher correlation coefficient in 10-fold cross-validation was obtained for linear regression (49.1% for sentence units, 49.91% for clause units), therefore linear regression was chosen for the final evaluation.

5.5 Coreference-related feature importance

An additional experiment was conducted to find out the correlation of coreference-related features with the gold unit scores. For that purpose, non-coreference-related features were removed from the clause unit dataset and then default WEKA's supervised attribute selection was performed. It resulted in choosing the following features:

- count of all mentions inside the unit (0.7284),
- count of all coreference clusters with at least one mention inside the unit (0.5575),
- average character length of all mentions inside the unit (0.1374),
- the mention count of the largest coreference cluster with at least one mention inside the unit (0.0217)
- the character length of the longest mention inside the unit (0.0212),
- average word length of unit mentions divided by the unit character length (-0.0812),
- total number of mentions of all clusters having at least one mention inside the unit (-1.0208).

Linear regression model trained using these features yielded weights presented in parentheses. This shows that an informative unit should:

- have large number of long (in terms of characters) mentions, but not too long (in terms of words) compared to the unit character size,
- contain an element from a large cluster,
- have elements from many clusters, yet not many large clusters.

Similar results were obtained for the sentence unit setting.

6 Evaluation

There is no consensus in the scientific environment about the single best metric used to evaluate single-document summarization systems. One of the most simple and popular metric is *ROUGE* [44, 45, 46]. The acronym is expanded as *Recall-Oriented Understudy for Gisting Evaluation* and covers several metrics:

- $ROUGE_n$, which counts n-gram co-occurrences between reference (*gold*) summaries and system summary,
- *ROUGE-L*, which searches for longest common subsequences of words between reference and system summaries,
- *ROUGE-W*, which is a version of *ROUGE-L*, weighing longest common subsequences by the lengths of discontinuous fragments in them,
- *ROUGE-S*, which uses so called *skip-bigrams*, i.e. pairs of words, possibly divided by other words,
- *ROUGE-SU*, which is an extension of *ROUGE-S*, utilising unigrams too.

In this article, I have chosen $ROUGE_n$ to be the metric used for evaluation. It is both the simplest one and also the one shown to well correlate with human assessment of summary quality, especially $ROUGE_2$ for single document summarization and $ROUGE_1$ for very short summaries [46]. $ROUGE_n$ is calculated in the following way:

$$ROUGE_n = \frac{\sum_{S \in RS} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in RS} \sum_{gram_n \in S} Count(gram_n)},$$

where:

- n – n-gram word count,
- RS – a set of reference (gold) summaries,
- $Count_{match}(gram_n)$ – the number of occurrences of given n-gram both in a reference summary and an automatic (system) one,
- $Count(gram_n)$ – the number of occurrences of given n-gram in a reference summary.

The denominator contains the total number of occurrences of all n-grams in all reference summaries, while the numerator has the number of n-grams in the automatic summary, which do map to occurrences in a reference summary.

The numerator clearly indicates, that $ROUGE_n$ is a recall-oriented measure, not a precision-oriented one, as its sibling *BLEU* [47], a measure for machine translation evaluation.

Evaluation tool was written in Java, its source code is publicly available at Git repository: `http://git.nlp.ipipan.waw.pl/summarization/eval` (tag: `1.0`). For the purpose of obtaining n-grams, a simple tokenizer was implemented, splitting character stream on every sequence of non-alphanumeric characters. Example of such tokenization is given below:

Wyrażał przekonanie, że "złoty pociąg" istnieje "na ponad 99 procent".
[Wyrażał] [przekonanie] [że] [złoty] [pociąg] [istnieje] [na] [ponad] [99] [procent]

Moreover, after tokenization all tokens were mapped to lowercase, to encourage summarization systems do changes on sentence boundaries, possibly changing the case of letters at the beginning of original sentences.

Some researchers claim that there multiple gold summaries should not be used to create single reference summary, but allow a system to compare itself with one of the gold summaries it is most similar to. Therefore evaluation tool also facilitates an option to calculate *ROUGE* score of a system for each text using a single manual summary which gives the highest score as a reference. In such case we have:

$$ROUGE\text{-}M_n = \max_{S \in RS} \frac{\sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{gram_n \in S} Count(gram_n)}$$

This metric is included in the evaluation.

6.1 Comparison settings

Systems, to which I compared my summarization tool, are:

- Lakon – presented in Section 2.2,
- Świetlicka – presented in Section 2.2,
- OpenText – presented in Section 2.2,
- Baseline – system taking the first words from the original document to fill desired ratio.

To produce summaries of the first three systems, I used Multiservice. I created an application, which sends each original document from Polish Summaries Corpus to Multiservice to obtain a summary for each of three tested ratios:

- 20 percent,
- 10 percent,
- 5 percent.

Such approach needed some tuning, which details are shown in next section.

6.2 Obtaining comparative summaries

All systems implemented in Multiservice accept a single `ratio` parameter. Most of them understand it as the desired summary ratio in terms of sentence count. Initial approach to gather automatic summaries for these systems showed that this may result in large differences in output word counts of summaries

produced by various systems, as presented in box plots in Figure 1. The upper and lower "hinges" correspond to the first and third quartiles. The upper whisker extends from the hinge to the highest value that is within 1.5 * IQR of the hinge, where IQR is the inter-quartile range, i.e. the distance between the first and third quartiles. The lower whisker extends from the hinge to the lowest value within 1.5 * IQR of the hinge. Data beyond the end of the whiskers are outliers and plotted as blue crosses.

For Świetlicka sometimes summaries reached 40% word ratio, when 20% was requested. On the other hand, Lakon tended to select sentences shorter than average, most often outputting summaries shorter than requested. Baseline system of course was perfect in this area.

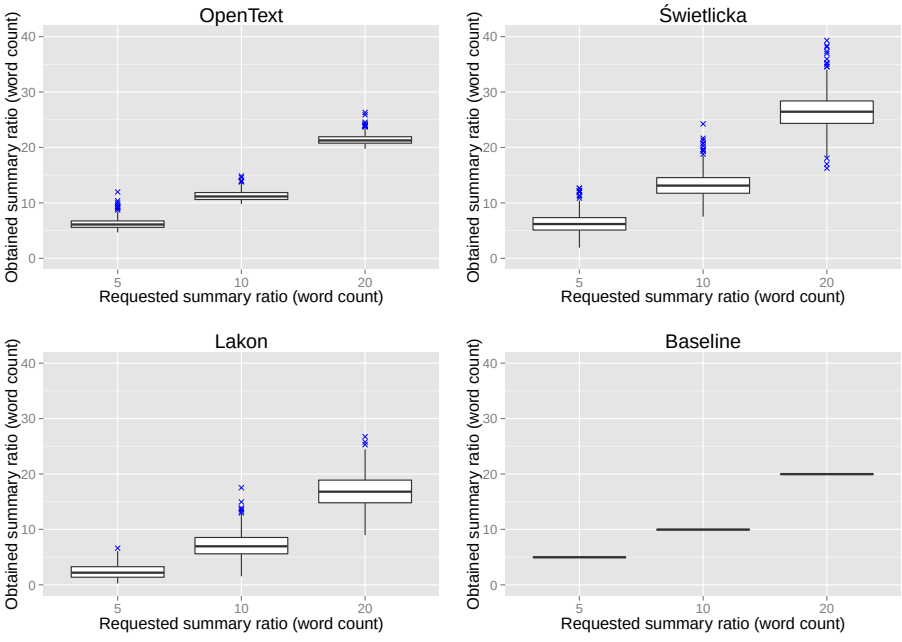

Fig. 1. Summary lengths with initial `ratio` use

However, taking into account our evaluation scheme it would be unfair to allow competing systems to produce summaries with such different word counts – as the evaluation concerns n-gram co-occurrences, the system with a summary with higher number of words can get a higher score. Another drawback of a constraint with only target sentence count is that the output summary length in terms of display space may vary very much. This may limit hypothetical practical use of a summarization system - it would be better to have a well-predictable space taken by a summary on screen.

Because of that, I decided to try changing requested ratio, if the output was too far from the desired length. The procedure was as follows:

1. Initially set the following parameters:
   - *ratio* to desired ratio (20, 10 or 5),
   - *iteration* to 1,
   - *prevDifference* to 100.
2. Send a MULTISERVICE request with *ratio* parameter and obtain a summary.
3. If the summary word length is greater than expected, decrease *ratio* parameter, otherwise increase *ratio* parameter.
4. Calculate the *difference* obtained and desired summary ratio in terms of word count.
5. End the procedure if any one condition is true:
   - $iteration > 20$,
   - $ratio < 1$,
   - $ratio > 99$,
   - $3 > difference > prevDifference$.
6. Increase *iteration* parameter.
7. Store *difference* into *prevDifference*.
8. Go to step 2.

The resulting summary was the one from penultimate iteration. The only unintuitive element of the algorithm is the fact that we do not end the procedure simply when in current iteration we obtained summary ratio worse than in previous iteration. There is an additional check of the difference from desired ratio being sufficiently small (smaller than 3). This is needed, because summarizers sometimes produce summaries longer in terms of word count, even when requested shorter summaries in terms of sentence count. In such case, we could be still far from desired word ratio and end the procedure.

The output of such approach was much closer to desired, as can be seen in Figure 2. This time BASELINE was replaced with EMILY to show, that the problem of too long summaries did not affect this summarizer (there was no iterative process in EMILY's case), just a single request with desired ratio). As the summaries are now of similar lengths, we may safely compare their quality.

6.3 Final results

Final results were collected via evaluation on test set, presented in Section 4.3. The set of reference (gold) summaries included only abstractive summaries, i.e. 5 summaries for each text. Results of evaluation using *ROUGE-N* and *ROUGE-M-N* for N=1,2,3 for ratios of 5%, 10% and 20% are presented in Tables 2, 3 and 4 respectively (see Section 6 for definition of evaluation metrics). Best results are marked with bold font.

To evaluate statistical significance of the differences between competing systems, one-sided t-test without the assumption of equal variances was used[3].

[3] The implementation details available at: `http://commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math3/stat/inference/TTest.html`

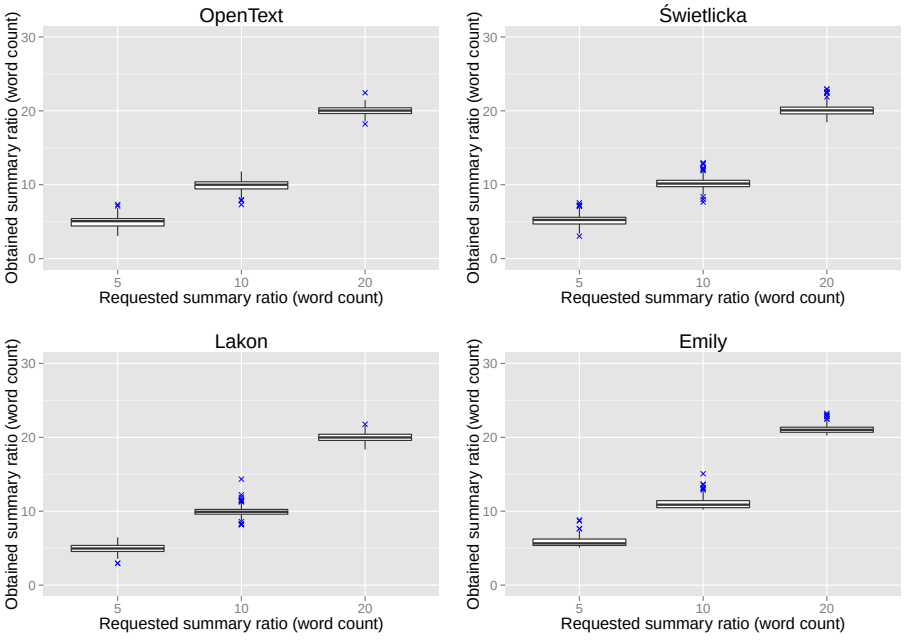

Fig. 2. Summary lengths with corrected `ratio` use

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-M-1	ROUGE-M-2	ROUGE-M-3
Baseline	**0.359** (0.119)	**0.170** (0.115)	**0.132** (0.107)	**0.533** (0.198)	**0.391** (0.252)	**0.355** (0.261)
Lakon	0.323 (0.093)	0.134 (0.078)	0.097 (0.071)	0.453 (0.136)	0.289 (0.162)	0.249 (0.165)
OpenText	0.248 (0.092)	0.047 (0.061)	0.028 (0.049)	0.340 (0.121)	0.114 (0.138)	0.085 (0.133)
Emily-C	0.270 (0.095)	0.069 (0.078)	0.048 (0.067)	0.359 (0.124)	0.148 (0.137)	0.118 (0.132)
Emily-S	0.272 (0.095)	0.064 (0.077)	0.042 (0.066)	0.364 (0.132)	0.145 (0.159)	0.113 (0.156)
Świetlicka	0.327 (0.100)	0.121 (0.079)	0.085 (0.069)	0.447 (0.137)	0.257 (0.161)	0.214 (0.162)

Table 2. Results for 5% ratio

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-M-1	ROUGE-M-2	ROUGE-M-3
Baseline	**0.448** (0.090)	**0.182** (0.087)	**0.139** (0.081)	**0.585** (0.131)	**0.380** (0.187)	**0.340** (0.196)
Lakon	0.417 (0.089)	0.147 (0.065)	0.103 (0.057)	0.509 (0.101)	0.270 (0.119)	0.225 (0.121)
OpenText	0.365 (0.092)	0.073 (0.056)	0.045 (0.043)	0.446 (0.103)	0.153 (0.106)	0.119 (0.100)
Emily-C	0.384 (0.082)	0.089 (0.058)	0.057 (0.049)	0.458 (0.088)	0.168 (0.098)	0.131 (0.095)
Emily-S	0.385 (0.080)	0.087 (0.059)	0.056 (0.050)	0.454 (0.090)	0.165 (0.101)	0.130 (0.101)
Świetlicka	0.442 (0.088)	0.155 (0.067)	0.107 (0.057)	0.539 (0.101)	0.280 (0.117)	0.231 (0.116)

Table 3. Results for 10% ratio

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-M-1	ROUGE-M-2	ROUGE-M-3
Baseline	0.558 (0.077)	0.214 (0.076)	0.160 (0.071)	**0.657** (0.091)	**0.383** (0.142)	**0.339** (0.149)
Lakon	0.554 (0.077)	0.208 (0.067)	0.144 (0.054)	0.629 (0.080)	0.333 (0.104)	0.274 (0.101)
OpenText	0.513 (0.082)	0.136 (0.063)	0.090 (0.050)	0.585 (0.086)	0.225 (0.097)	0.179 (0.090)
Emily-C	0.529 (0.076)	0.151 (0.053)	0.100 (0.041)	0.588 (0.078)	0.242 (0.080)	0.192 (0.076)
Emily-S	0.530 (0.072)	0.154 (0.056)	0.104 (0.047)	0.594 (0.073)	0.247 (0.083)	0.201 (0.081)
Świetlicka	**0.580** (0.081)	**0.226** (0.070)	**0.161** (0.059)	0.654 (0.085)	0.358 (0.104)	0.298 (0.101)

Table 4. Results for 20% ratio

However, only for the largest summary ratio the assumption of normality (tested with Kolmogorov–Smirnov test[4]) was confirmed for most systems and metrics (34 out of 36, which is acceptable at 0.05 significance level). In 5% and 10% summarization ratio the normality assumption was only fulfilled for 20 and 22 metric-system combinations, respectively. Therefore Table 5 shows the significantly worse systems for each system and metric only of 20% ratio setting.

System	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-M-1	ROUGE-M-2	ROUGE-M-3
1. Baseline	3, 4, 5	3, 4, 5	2, 3, 4, 5	2, 3, 4, 5	2, 3, 4, 5, 6	2, 3, 4, 5, 6
2. Lakon	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5
3. OpenText						
4. Emily-C	3	3	3		3	
5. Emily-S	3	3	3		3	3
6. Świetlicka	1, 2, 3, 4, 5	2, 3, 4, 5	2, 3, 4, 5	2, 3, 4, 5	2, 3, 4, 5	2, 3, 4, 5

Table 5. Statistically significant differences for 20% ratio

The results are not encouraging: Emily manages to significantly outperform only OpenTextSummarizer. The sentence unit version (Emily-S) scores a little better than the clause unit version (Emily-C), however the results are much worse than these obtained by the Świetlicka system and the Baseline. It is important to notice that the Baseline is very strong: only Świetlicka for ROUGE-1 metric is able to get higher score. The results are even worse for 5% and 10% ratios – the Baseline clearly wins for each metric in these cases.

This may indicate that a promising approach to summarize texts from the Polish Summaries Corpus is to start with the Baseline solution. Next step of a summarization algorithm may consist of searching for possible modifications to the Baseline.

7 Conclusions and Future Work

In this article I proposed a way to use multiple gold extractive summaries to score informative value of units in the original texts. I also introduced a machine

[4] The implementation details available at: `https://commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math3/stat/inference/KolmogorovSmirnovTest.html`

learning system able to create extractive summaries relying mostly on coreference features. Clause- and sentence-based extractive summarization methods were compared to existing summarization systems for Polish. All systems were for the first time thoroughly evaluated on a large corpus for various summarization ratios, showing the best performance of Świetlicka's solution and trivial Baseline.

Possible future work may involve further development of features or transformation and combinations of existing ones. It is necessary to test more advanced machine learning algorithms and most importantly compare the results on gold coreference data, because advanced preprocessing used by Emily may introduce many errors. Another approach would be to develop a non-greedy method (i.e. not ranking units independently, but rather taking into account their various subsets) for unit selection utilising coreference information.

Acknowledgements

The study is cofounded by the European Union from resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00.

References

[1] Ogrodniczuk, M., Głowińska, K., Kopeć, M., Savary, A., Zawisławska, M.: Coreference: Annotation, Resolution and Evaluation in Polish. Walter de Gruyter Inc. (2014)
[2] Broda, B., Burdka, Ł., Maziarz, M.: IKAR: An Improved Kit for Anaphora Resolution for Polish. In: COLING (Demos). (2012) 25–32
[3] Kopeć, M., Ogrodniczuk, M.: Creating a coreference resolution system for Polish. [48] 192–195
[4] Ogrodniczuk, M., Kopeć, M.: End-to-end coreference resolution baseline system for Polish. In Vetulani, Z., ed.: Proceedings of the Fifth Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznań, Poland (2011) 167–171
[5] Ogrodniczuk, M., Kopeć, M.: Rule-based coreference resolution module for Polish. In: Proceedings of the 8th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2011), Faro, Portugal (2011) 191–200
[6] Ogrodniczuk, M., Kopeć, M.: The Polish Summaries Corpus. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S., eds.: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavík, Iceland, European Language Resources Association (2014) 3712–3715
[7] Kopeć, M.: Zero subject detection for Polish. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers, Gothenburg, Sweden, Association for Computational Linguistics (2014) 221–225

[8] Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research and Development **2** (1958) 159–165
[9] Edmundson, H.P.: New methods in automatic extracting. J. ACM **16**(2) (April 1969) 264–285
[10] Marcu, D.: The automatic construction of large-scale corpora for summarization research. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '99, New York, NY, USA, ACM (1999) 137–144
[11] Erkan, G., Radev, D.R.: Lexpagerank: Prestige in multi-document text summarization. In: EMNLP, Barcelona, Spain (2004)
[12] Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization. (1997) 10–17
[13] Baldwin, B., Morton, T.S.: Dynamic coreference-based summarization. In: Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3). (1998)
[14] Azzam, S., Humphreys, K., Gaizauskas, R.: Using coreference chains for text summarization. In: Proceedings of the Workshop on Coreference and Its Applications. CorefApp '99, Stroudsburg, PA, USA, Association for Computational Linguistics (1999) 77–84
[15] Stuckardt, R.: Coreference-based summarization and question answering: a case for high precision anaphor resolution. In: International Symposium on Reference Resolution. (2003)
[16] Witte, R., Bergler, S.: Fuzzy Coreference Resolution for Summarization. In: Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS), Venice, Italy, Università Ca' Foscari (June 2003) 43–50 `http://rene-witte.net`.
[17] Bergler, S., Witte, R., Khalife, M., Li, Z., Rudzicz, F.: Using Knowledge-poor Coreference Resolution for Text Summarization. In: Workshop on Text Summarization. Document Understanding Conference (DUC), Edmonton, Canada, NIST (May 2003) `http://duc.nist.gov/`.
[18] Steinberger, J., Poesio, M., Kabadjov, M.A., Jeek, K.: Two uses of anaphora resolution in summarization. Inf. Process. Manage. **43**(6) (November 2007) 1663–1680
[19] Hendrickx, I., Bouma, G., Coppens, F., Daelemans, W., Hoste, V., Kloosterman, G., Mineur, A.M., Van Der Vloet, J., Verschelde, J.L.: A coreference corpus and resolution system for dutch. In Calzolari, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Tapias, D., eds.: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, European Language Resources Association (ELRA) (may 2008) http://www.lrec-conf.org/proceedings/lrec2008/.
[20] Kaplan, D., Iida, R., Tokunaga, T.: Automatic extraction of citation contexts for research paper summarization: A coreference-chain based approach. In: Proceedings of the 2009 Workshop on Text and Citation Analy-

sis for Scholarly Digital Libraries. NLPIR4DL '09, Stroudsburg, PA, USA, ACL (2009) 88–95
[21] Smith, C., Danielsson, H., Jönsson, A.: A more cohesive summarizer. In: Proceedings of COLING 2012: Posters, The COLING Organizing Committee (2012) 1161–1170
[22] Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proceedings of EMNLP-04 and the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2004)
[23] Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer networks and ISDN systems **30**(1) (1998) 107–117
[24] Mitkov, R., Evans, R., Orăsan, C., Dornescu, I., Rios, M.: Coreference resolution: To what extent does it help NLP applications? In Sojka, P., Horák, A., Kopeček, I., Pala, K., eds.: Text, Speech and Dialogue - 15th International Conference, TSD 2012, Brno, Czech Republic, September 3-7, 2012. Proceedings. Volume 7499 of Lecture Notes in Artificial Intelligence., Heidelberg, Springer-Verlag (2012) 16–27
[25] Orăsan, C.: Comparative evaluation of term-weighting methods for automatic summarization. Journal of Quantitative Linguistics **16**(1) (2009) 67–95
[26] Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A.: BART: A modular toolkit for coreference resolution. In: Association for Computational Linguistics (ACL) Demo Session. (2008)
[27] Dudczak, A.: Metody maszynowego uczenia w automatycznym streszczaniu tekstów. Master's thesis, Poznan University of Technology, Poland (2007)
[28] Dudczak, A., Stefanowski, J., Weiss, D.: Automatyczna selekcja zdań dla tekstów prasowych w języku polskim. Technical Report RA-03/08, Institute of Computing Science, Poznan University of Technology, Poland (2008)
[29] Dudczak, A., Stefanowski, J., Weiss, D.: Comparing performance of text summarization methods on polish news articles. In: Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference, Zakopane, Poland (2008) Available on-line.
[30] Dudczak, A., Stefanowski, J., Weiss, D.: Evaluation of sentence-selection text summarization methods on polish news articles. Foundations of Computing and Decision Sciences **1**(35) (2010) 27–41
[31] Woliński, M.: Morfeusz – a practical tool for the morphological analysis of Polish. In Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K., eds.: Proceedings of the International Intelligent Information Systems: Intelligent Information Processing and Web Mining'06 Conference, Wisła, Poland (June 2006) 511–520
[32] Świetlicka, J.: Metody maszynowego uczenia w automatycznym streszczaniu tekstów. Master's thesis, University of Warsaw, Poland (2010)
[33] Piasecki, M.: Polish tagger TaKIPI: Rule based construction and optimisation. Task Quarterly **11**(1–2) (2007) 151–167
[34] Przepiórkowski, A.: The IPI PAN Corpus: Preliminary version. Institute of Computer Science, Polish Academy of Sciences, Warsaw (2004)

[35] Rotem, N.: Open text summarizer (2003)
[36] Weiss, D.: Korpus Rzeczpospolitej. [on-line] `http://www.cs.put.poznan.pl/dweiss/rzeczpospolita` (2002)
[37] Nenkova, A., Passonneau, R.J., McKeown, K.: The pyramid method: Incorporating human content selection variation in summarization evaluation. TSLP **4**(2) (2007) 16–44
[38] Waszczuk, J.: Harnessing the crf complexity with domain-specific constraints. the case of morphosyntactic tagging of a highly inflected language. In: COLING. (2012) 2789–2804
[39] Radziszewski, A., Śniatowski, T.: Maca — a configurable tool to integrate Polish morphological data. In: Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation. (2011)
[40] Przepiórkowski, A., Buczyński, A.: Spejd: Shallow Parsing and Disambiguation Engine. In Vetulani, Z., ed.: Proceedings of the 3rd Language & Technology Conference, Poznań, Poland (2007) 340–344
[41] Waszczuk, J., Głowińska, K., Savary, A., Przepiórkowski, A.: 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 (2010) 531–539 PTI.
[42] Ogrodniczuk, M., Lenart, M.: Web Service integration platform for Polish linguistic resources. [48] 1164–1168
[43] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. **11**(1) (2009) 10–18
[44] Lin, C.Y., Hovy, E.: Manual and automatic evaluation of summaries. In: Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4. AS '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 45–51
[45] Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. NAACL '03, Stroudsburg, PA, USA, Association for Computational Linguistics (2003) 71–78
[46] Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, Association for Computational Linguistics (2004) 74–81
[47] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 311–318
[48] Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., eds.: Proceedings of the Eight International

Conference on Language Resources and Evaluation (LREC'12). In Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., eds.: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, European Language Resources Association (ELRA) (May 2012)