ON SUMMARIZATION SUPPORTING READABILITY AND TRANSLATABILITY

Sheremetyeva S. O. (lanaconsult@mail.dk)

LanA Consulting ApS, Copenhagen, Denmark
National Research South Ural State University, Chelyabinsk, Russia

The article describes a methodology of developing an interactive computer system for supporting a single document text-to-text summarization process focusing on providing for high readability and translatability of the generated summary that, in turn, facilitates further human or automatic processing of the summary text, translation being the most important. The decisions on content selection is delegated to a human but are largely supported by the system. High readability and translatability of the generated text is provided by controlling the syntax of the nascent summary. The approach is a combination of empirical and rational NLP techniques and incorporates a language independent algorithm and language-dependent knowledge base. The validity of the approach was proved by its implementation into a summarizer for scientific papers in the domain of mathematical modelling in the Russian language. The summarizer is fully operational. The methodology presented in this paper is highly portable and allows for extending the summarizer to other domains and languages.

Keywords: summarization, readability, translatability, machine processing, interactivity

1. Introduction

Development of efficient summarization systems is of special importance for scholars and developers. With the cumulative total, which is estimated to pass 50 million scientific papers [9] researchers can only keep abreast of the growing number of scientific developments through summary digest. While it is generally recognized that summaries should be created operatively and contain all the key content of a full document, such properties of a summary as readability and translatability often escape both the authors and summary systems developers' attention. Good readability means that a text is easy to understand for a human, especially for a human translator who, as a rule, does not possess enough of domain knowledge and needs to clearly understand the syntactic dependences of the original text. High translatability means that a summary can be well "understood" by a machine translation system as translators and especially the authors, more and more nowadays use MT tools. It is not uncommon to see summary texts that are very problematic both for humans and machines, such as the following:

(1) *Строятся и исследуются аналитическими методами математические модели напряженных состояний тонкостенных цилиндрических оболочек, продольных, поперечных и спиральных менее прочных слоев (прослоек) в них, в том числе содержащих дефекты, более прочных слоев с дефектами, при нагружении оболочек внутренним давлением и осевой силой.*

While the terminology problem can be solved with correct terminological lexicons, it is such linguistic phenomena as long sentences, distant dependencies, complex syntactic structures, coordination, etc., that lower the levels of texts readability and translatability [21].

In this paper we describe our effort to develop a methodology for a real life tool that could support high readability and translatability in the summarization process. The approach is a combination of empirical and rational natural language processing techniques including interactive computer-supported content elicitation and fully automatic text generation. The methodology is realized in a summarization tool for research papers on mathematical modelling in the Russian language.

2. Related work and motivation

Major summarization strategies fall into extraction, abstraction or mixed paradigms and are realized in the frame of linguistic, statistical or hybrid approaches.

In linguistic approaches text summarization is considered as a transformation process by which knowledge representation structures, as generated by a natural language processing (NLP) system, are mapped into conceptually more abstract, condensed knowledge structures that account for document content[7,11]. Knowledge representation structures such as, *Schemata*, [16], *Rich Semantic Graphs* [6] or *predicate-argument representations* [18], to name just a few, are then transformed into a summary text by natural language generation (NLG) techniques. Linguistic approaches are normally abstracting and can provide higher quality summaries, but they are knowledge–heavy and suffer coverage problem. In pure statistical, normally extracting, summarization the most relevant sentences of a document are extracted according to a certain statistical metrics, e.g., a classical tf-idf term weighting scheme, and inserted in the summary as such [1, 15, 20]. Statistical systems though providing for coverage can still produce problematic output. Most popular now are mixed-paradigm hybrid approaches that on top of statistical calculations rely on a shallow to deep knowledge bases and more or less interwoven extraction and transformation components. The latter modifies selected fragments with one or several of the following techniques,—paraphrase induction [22], fragment recycling [5], sentence simplification [24], sentence compression [4], sentence fusion [2, 3] or predicting the selection of lexical units and their position in the summary [15,17]. The knowledge base in such systems can contain from lists of predicates, to rhetorical and sentence templates [23]. Summary-relevant fragments in the input are normally spotted by building conditional models over some statistical features, for example, commonly occurring word co-occurrences in summary sentences [14, 23] or words across adjacent sentences in particular semantic roles [2,10].

We aim at developing a real world application for authors and editors that do not only guarantee the correct content of a summary but also takes care of the summary text structure, making it highly readable and translatable. So far we were not able to find any research which would combine text compression with readability/translatability issues. However, now, when summaries are the priority types of documents that undergo further processing (e.g., human or machine translation) readability and translatability are of great importance. This is especially relevant for the Russian language, whose rich morphology and free word order allow though grammatically correct, but still low readable/translatable long, syntactically extremely complex and ambiguous sentences. The authors trying to make their summaries more informative and much compressed often abuse these features of Russian that causes a lot of problems in further processing of the documents. Instructions only, and simple style checkers that are not so far sufficiently developed, especially for Russian, do not serve the purpose. Let alone that a good style does not always guarantee high translatability. It actually can be the other way around. Analysis of research in automation of text processing tasks shows that machines are better at syntax than at semantics, and systems, in which the division of labour is "semantics for humans, syntax for the computer" allow for much better results [12,13]. We therefore make our system interactive and delegate content selection to the user, while (unlike style checkers) providing extensive linguistic support and full automation of text restructuring based on natural language generation techniques. The knowledge and processing rules in our approach as in other most promising research [8, 22, 23] draw heavily on domain restrictions that contribute a lot to the system viability.

3. Approach Overview

The overall architecture of the summarizer is shown in Figure 1. The system consists of

- Domain-tuned knowledge base
- automatic noun phrase (NP) extractor, NP and predicate phrase (VP) chunkers
- interactive content elicitation module
- automatic content representation module
- automatic summary text generator
- user-friendly interface

Human intervention occurs at the stage of content selection and is linguistically supported by (i) alerting the user about the content of an initial document with explicitly marked noun and predicate terminology (see Fig. 2) and (ii) providing lexical menus and knowledge elicitation templates. The results of knowledge elicitation are further automatically processed into an underlying representation followed by the summary generation in a syntactically controlled language providing for high readability and translatability of the summary. The approach is of hybrid nature and includes language- and domain-independent algorithms that run over language-dependent domain-tuned lexical knowledge. This makes the methodology portable across domains and languages. The details are further described on the example of a summarizer for the domain of mathematical modelling in the Russian language.

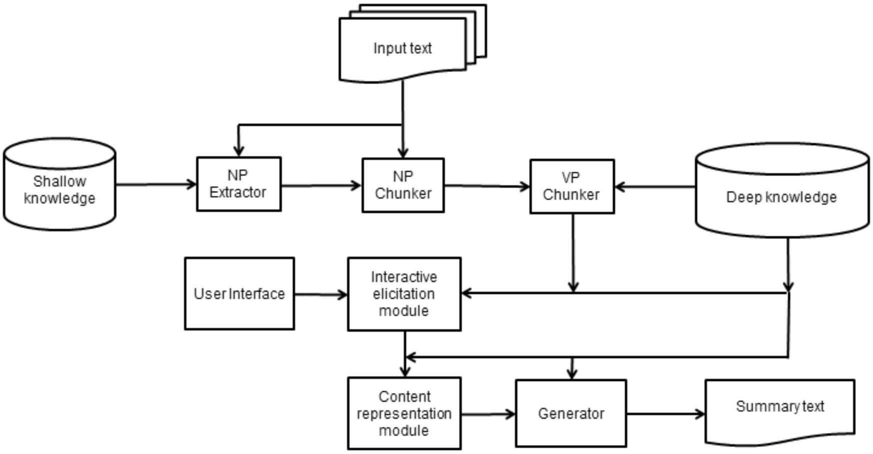


Fig. 1. An overall architecture of the summarizer

4. Knowledge

The summarizer knowledge builds on mathematical modelling corpus analysis and consists of application specific lexicons and processing rules. It is organized in blocks used by different system modules.

Block 1 (shallow knowledge) is used by the NP extractor and chunker. The NP extractor knowledge consists of a number of shallow lexicons and deletion rules; the lexicons are parts-of-speech-sorted lists of wordforms from the corpus that are forbidden in certain positions in Russian NPs; the rules draw on NP structural restrictions. The extractor was ported from English and tuned to the domain in question; see details in [19].

Block 2 (deep knowledge) includes an information-rich lexicon of predicates and rules to handle predicate knowledge in the analysis and generation modules of the summarizer. The lexicon is composed of a set of single sense entries defined as follows:

dictionary::= {entry}+
entry::= part-of-speech, typed morphological features
(number, gender, time, etc.),
domain relevant morphological forms explicitly listed in the entry
case-roles, linear patterns

The set of parameters (or fields) for predicate specification in the entries of the lexicon is strictly determined by the needs of application and corpus analysis. Explicitly listed relevant morphological forms of the predicates help their identification in the input text. Certain conventions are used in organizing knowledge in the lexicon that facilitates processing. For example, passive and active forms of a predicate are considered to be realizations of different lexemes.

The maximum set of the predicate case-roles includes *subject, direct object, indirect object1, manner, place, means, purpose, time*, *source, destination, condition, agent* and *other.* Linear patterns code information about the order of realization of case-roles of certain semantic status in the syntactic structure of the summary sentence. The knowledge base is further augmented with content elicitation templates built on the predicate knowledge and so as to prevent generation of summary sentences with complex syntactic structures. The templates are empty predicate—argument frames. Each slot in the elicitation template contains a main (predicate) slot and slots corresponding to the particular predicate case-roles. The domain restrictions are reflected in the corpus-based predicate vocabulary and predicate morphological, syntactic and semantic properties (case-role sets and patterns).

5. Workflow

The workflow in the summarizer is as follows:

Input markup. The goals of this automatic analysis stage is to (i) alert the user on the paper terminology, (ii) automate user manipulations with the text chunks and (iii) link predicate lexemes with the deep knowledge in the predicate lexicon, elicitation templates and processing rules in the system knowledge base. The document first goes to the automatic extractor, which produces a list of one- to four component noun phrases. The output NP list is then supplied to a shallow analyzer component, which by matching the extracted list against the input text from left to right chunks NP terminology in the document. The remaining text is supplied to another analyzer component, which matches it against the strings in the morphological zones of the predicate lexicon. In case of a match the text string is chunked as a predicate and linked to all the information of the corresponding predicate entry. The fact that the predicate chunker does not run within chunked NPs practically lifts the ambiguity problem in predicate identification. Finally the document is turned into an interactive ("clickable") text with NPs and VPs highlighted and presented to the user.

Morphological normalization of text predicates. The goal of this stage is to create a menu of predicates in normalized forms to facilitate content selection. In our system we normalize text forms of the predicates to their finite forms while keeping the features of time, voice, aspect, gender and number of these lexemes as used in the text. For example, if the text form of a predicate is "*разделяющего* " ("*separating*") with the morphological features of *present participle, masculine, singular, genitive, active voice*, it is stored in the predicate menu in the form of "*разделяет*" with morphological features of *active finite form, present tense, masculine, singular, genitive.* This makes it easy for the user to see the lexeme as a syntactic predicate of a summary sentence. Predicate normalization rules work over the knowledge stored in the morphological zone of the lexicon.

[1] These are just case-role labels to show that their fillers occupy positions of subject, direct and indirect objects in the syntactic structure of a text sentence.

Elicitation of the summary content. One goal of this interactive stage is to elicit from the user the knowledge about the summary content while controlling the syntactic structure of the nascent summary. Another goal of this stage is to facilitate building internal representation of the elicited content. In fact, the user is implicitly prompted to decompose a complex input text into predicate phrases by supplying text strings into predicate templates from the system knowledge base. The elicitation procedure halts when all the content for the summary is elicited.

Internal content representation. Once the user fills the appropriate slots of a predicate template, the system automatically produces the internal representation of an elicited quantum of the summary content as follows

(2) ***(P 2 “(V_pres_3prs_sg) “является*** */ is”*
(1 subj “данный алгоритм */the given algorithm”)*
(4indir-obj “основной частью итерационного метода решения задачи сильной сходимости */main part of the iteration method for solving the strong convergence problem”)*
(5place “в распознавании образов” */ in image recognition))]*

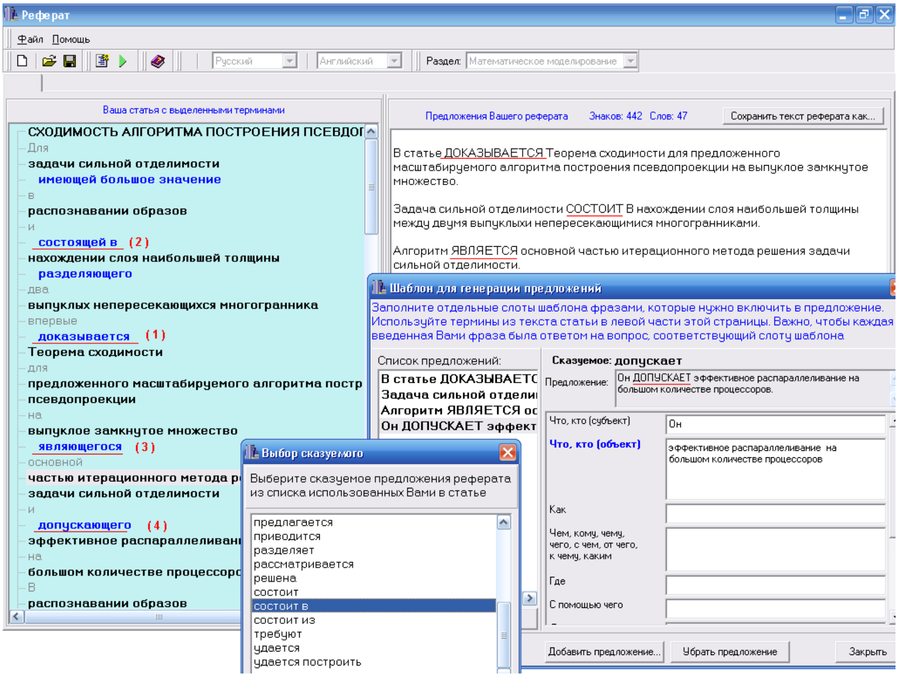
The output of this stage is a set of filled predicate templates.

Generation of the summary text. The content representation is passed to the text planner that treats individual predicate templates as representations of summary sentences, while case-role fillers are treated as sentence constituents. By default the planner orders a set of predicate-argument structures following the order of their creation by the user. This order can be changed interactively through the user interface. The order of case-role fillers as sentence constituents (that are treated as blocks without any analysis) is defined according to the corresponding linear pattern in the predicate entry in the lexicon. For example, the internal representation of the Russian predicate P2 *“является/is”* shown in (2) will be linearized according to the pattern (5 1 × 4) stored in the lexicon entry of this predicate:

(3) ***(****5place****: “в распознавании образов*** *” /in image recognition)* ***(****1 subj “****данный алгоритм*** */the given algorithm”)* ***x : “являетсяся*** */ is”* ***(****4indir-obj “****основной частью итерационного метода решения задачи сильной сходимости*** */ the main part of the iteration method for solving the strong convergence problem”)*

6. Implementation

The summarization methodology is implemented it into a program,—a text-to-text interactive summarizer for the domain of scientific papers on mathematical modelling in the Russian language. The programming is done in C++ for the Windows operational environment. Using common graphical user interface tools (such as dialogue boxes, menus, templates, etc.), the system guides the user through the paces of content selection and creation of a highly readable and translatable summary.


Реферат
Файл Помощь
Русский
Английский
Раздел: Математическое моделирование
Ваша статья с выделенными терминами
Предложения Вашего реферата
Знаков: 442 Слов: 47
Сохранить текст реферата как...
Шаблон для генерации предложений
Выбор сказуемого
Добавить предложение...
Убрать предложение
Закрыть

СХОДИМОСТЬ АЛГОРИТМА ПОСТРОЕНИЯ ПСЕВДОГ
Для
задачи сильной отделимости
имеющей большое значение
в
распознавании образов
и
состоящей в (2)
нахождении слоя наибольшей толщины
разделяющего
два
выпуклых непересекающихся многогранника
впервые
доказывается (1)
Теорема сходимости
для
предложенного масштабируемого алгоритма постр
псевдопроекции
на
выпуклое замкнутое множество
являющегося (3)
основной
частью итерационного метода р
задачи сильной отделимости
и
допускающего (4)
эффективное распараллеливан
на
большом количестве процессор
В
распознавании образов

В статье ДОКАЗЫВАЕТСЯ Теорема сходимости для предложенного масштабируемого алгоритма построения псевдопроекции на выпуклое замкнутое множество.

Задача сильной отделимости СОСТОИТ В нахождении слоя наибольшей толщины между двумя выпуклыхи непересекающимися многогранниками.

Алгоритм ЯВЛЯЕТСЯ основной частью итерационного метода решения задачи сильной отделимости.

Заполните отдельные слоты шаблона фразами, которые нужно включить в предложение. Используйте термины из текста статьи в левой части этой страницы. Важно, чтобы каждая введенная Вами фраза была ответом на вопрос, соответствующий слоту шаблона

Список предложений:
В статье ДОКАЗЫВАЕТС
Задача сильной отдели
Алгоритм ЯВЛЯЕТСЯ ос
Он ДОПУСКАЕТ эффект

Сказуемое: допускает

Предложение: Он ДОПУСКАЕТ эффективное распараллеливание на большом количестве процессоров.

Что, кто (субъект): Он
Что, кто (объект): эффективное распараллеливание на большом количестве процессоров
Как:
Чем, кому, чему, чего, с чем, от чего, к чему, каким:
Где:
С помощью чего:

Выберите сказуемое предложения реферата из списка использованных Вами в статье

предлагается
приводится
разделяет
рассматривается
решена
состоит
состоит в
состоит из
требуют
удается
удается построить

Fig. 2. A screenshot of a fragment of the summarizer interface in the process of content elicitation

The start up page of the interface displays two empty panes. On downloading a document the interface automatically displays the document text made interactive with noun and predicate phrases highlighted in different colours as shown in Fig. 2. Simultaneously the first elicitation template pops up completely or partially filled. The predicate slot is filled with the lexeme “*рассматривается/considered”;* the “subject” slot is filled with the most relevant term (the first NP from the title) and the “place” slot contains a fixed string *“в статье/in the paper”*. The user can accept, edit or delete the template or any of its fillers. Any text string can be transferred from the left pane into an elicitation template on a double click. Template slots that correspond to predicate case-roles are marked with “human” questions conveying the semantic status of the case-roles/slots (e.g.,” 5 place”à”где”/“where”; “ 3 manner” à”где”/“how”). The user is supposed to fill the slots with text strings that answer the questions. Immediately on the completion of filling a predicate template a summary sentence is generated on the back right pane for user control as shown in Fig. 2. The “Add a new sentence” button on the predicate template calls a predicate menu. A predicate selected in menu calls for the corresponding template to be filled. The pop-up predicate menu is provided with an empty type in area in case the user prefers to use a predicate other than one of those used in the document. A “new” typed in predicate calls a default predicate template. It is possible to call for a predicate template using a “short cut” by directly clicking on the highlighted predicate in interactive text. The Interface includes a spell checker and provides supporting information such as the number of characters and words of already generated sentences (shown on the top of the right pane). The "Save as text" button saves a summary in a text file. The "Save project" and "Open project" selections in the main File menu allow for saving/opening summary drafts to work on them in several takes.

7. Evaluation

The specificity of the approach is reflected in the aspects that undergone evaluation. Neither content selection, nor document compression evaluations are applicable. The content as selected by the user was considered to be fully correct. The level of document compression fully depends on the lengths of text chunks that the user supplies to predicate slots. The generator is responsible for the predicate morphological forms that should agree with the fillers in the subject slots and the order of realization of case-role slots fillers that are treated as text blocs without any change.

The evaluation was therefore done on the following four levels: grammaticality, readability, translatability and user satisfaction based on the expert judgment. The body of experts included translators, professors, and students from mathematical and linguistic departments of the South Ural State University, Russia (http://www.susu.ac.ru). All experts were native Russian speakers with good proficiency in English. The evaluation tasks were divided between the participating experts. As a starting point 200 full papers were divided between the participating experts—the mathematicians who were first asked to create summaries following instructions only, then with the help of existing on-line style checkers and at last they created summaries with the tool (to guarantee correct content). The on-line Russian style checkers were almost immediately rejected by the experts as completely useless for the task. Instructions-only summaries were also rejected by mathematicians as too "linguistic" or vice-versa 'too general" to help summary composition.

Grammaticality of the tool generated summaries was judged by the students—linguists. The sentences of the summary were considered grammatical if they obeyed the rules of the Russian grammar. The experts reported ~ 91% of grammatically correct sentences. Grammar mistakes were caused by the incomplete coverage of the predicate lexicon, when due to the lack of a predicate default templates (and linear patterns) were used which caused problems in word order.

Improvement in readability of the tool-generated summaries was judged by participating mathematicians and translators by comparing them with the original (author written summitries) ones. For translators (who are not mathematicians) readability means clear and unambiguous syntax. This evaluation was qualitative and both parties reported improvement in this parameter.

To evaluate translatability translators and students-linguists were asked to translate original and tool generated summaries into English with two most popular and best developed for Russian free online MT systems,—PROMT and GOOGLE TRANSLATE. The former is a rule-based MT system, while the latter is a statistical MT system. Terminology translation was excluded from the examination, as none of these online systems were supposed to cover properly the terminology of our domain. Syntactic mistakes that occurred due to wrong terminology translation were little in number and, therefore, neglected. First of all it was found that both RBMT (PROMT) and SMT (GOOGLE TRANSLATE) suffer, though differently, from the same linguistic phenomena in the source Russian texts. There was practically no author written summary which could provide for a correct MT. On the other hand, practically all tool generated summaries (terminology neglected) provided for correct MT.

Two groups of experts, who have worked with the system, authors (mathematicians) and translators reported high satisfaction with the summarization results and the simplicity of the knowledge elicitation procedure. It took around an hour of training for them to get acquainted with the system. They underline the usefulness of full document mark-up for the summarization procedure that made the input much more understandable. The quality of mark-up depends on the correctness of NP and VP chucking. The latter was evaluated by comparing tool-chunked NPs and VPs with the gold lists of these phrases (manually for every document). The results proved to be rather high due to the high quality performance of the NP extractor, rich morphology of the Russian language and tuning the system knowledge to the restricted domain (see Table 1).

Table 1. Chunking evaluation results

NP chunking (%)		**Predicate chunking (%)**		**Grammar (%)**
recall	precision	recall	precision	correctness
95	94	98	96	93

8. Conclusions

In this paper we described a methodology for developing a system for generating a single document text-to-text-summary that involves computer-supported human intervention at the stage of content selection and provides for high readability and translatability of the summary. The validity of the approach was proved by its implementation into a summarizer for scientific papers in the domain of mathematical modelling in the Russian language. The summarizer is fully operational.

The static knowledge sources including the shallow and deep lexicons as well as other analysis- and transfer-related knowledge blocks have been compiled based on the sublanguage analysis and provide for good coverage. The summarization algorithm is universal and robust as it excludes such statistically or NLP expensive techniques as combinatorial computations or tagging and parsing. The evaluation results presented in this paper confirm the viability of the approach.

We plan to extend this work in a number of ways. We are currently working on making our summarizer fully automatic.

Due to the universal metalanguage of knowledge representation and language-independent processing algorithms the methodology presented in this paper is highly portable and allows for extending the summarizer on other domains and languages. We intend just that. It takes tuning the shallow lexicons of the NP extractor to a new domain that within one language can cost a couple of one man days and update a lexicon of predicates, a lot of predicate knowledge already acquired being reused. The program shell and developer tools are completely reusable. One more perspective is to apply the techniques described to multilingual applications, e.g., multilingual search or machine translation.

References

1. *Alekseev A., Loukachevitch N.* (2012), Use of Multiple Features for Extracting Topics from News Clusters. Proc SYRCODIS'2012, pp. 3–11.
2. *Barzilay R. and Lillian Lee.* (2004), Catching the drift: Probabilistic content models, with applications to generation and summarization. In DanielMarcu Susan Dumais and Salim Roukos, editors, HLT-NAACL 2004: Main Proceedings, pages 113–120, Boston.
3. *Barzilay R. and Kathleen R. McKeown.* (2005), Sentence fusion for multidocument news summarization. Computational Linguistics, 31(3): 297–328.
4. *Clarke J. and Mirella Lapata.* (2007), Modelling compression with discourse constraints. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 1–11.
5. *Daumé H. III and Daniel Marcu.* (2005), Induction of word and phrase alignments for automatic document summarization. Computational Linguistics, 31(4): 505–530, December.
6. *Fathy Ibragim.* (2012), Rich semantic representation based approach for text generation. In Proc. The 8th International Conference on Informatics and Systems (INFOS). Cairo, Egypt 14–16 May.
7. *Hahn U., and Reimer U.* (2001), Knowledge-Based Text Summarization: Salience and Generation Operators for knowledge Base Abstraction. In Inderjeet Mani and Mark T. Maybury, editors, Advances in Automatic Text Summarization! Massacusets Institute of Technology, pp. 215–232.
8. *Khodra M. L., D. H. Widyantoro, E. A. Aziz, B. R. Trilaksono.* (2011), Free Model of Sentence Classifier for Automatic Extraction of Topic Sentences. Journal of ICT Research and Applications, Vol. 5C No. 1.
9. *Jinha A. E.* (2010), Article 50 Million: An Estimate of the Number of Scholarly Articles in Existence. Learned Publishing, 23 (3) (2010), pp. 258–263.
10. *Lapata M.*(2003), Probabilistic text structuring: Experiments with sentence ordering. In Proceedings of the 41st AnnualMeeting of the Association for Computational Linguistics, pages 545–552, Sapporo, Japan.
11. *Leontyeva N. N.* (2003), Semantic Dictionary for Text Understanding and Summarization // International Journal of Translation. V. 15. № 1. P. 107–114.
12. *Leuski A., Lin C-Y., Hovy E.* (2003), iNeATS: Interactive Multi-Document Summarization. In Proc. The 41st AnnualMeeting of the Association for Computational Linguistics, Sapporo, Japan.

13. *Lin, J., M. Nitin., B. Dorr.* (2010), Putting the User in the Loop: Interactive Maximal Marginal Relevance for Query-Focused Summarization. In Proceedings of the The 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden, July 11–16.
14. *Lloret E. A.* (2009), Gradual Combination of Features for Building Automatic Summarisation Systems Text [Text] / E. Lloret, M. Palomar // Speech and Dialogue. — Heidelberg, — P. 16–23.
15. *Lukashevich N. V., Dobrov B. V.* (2009), Automatic annotation of a news cluster based on thematic representation [Avtomaticheskoye annotirovaniye novostnogo klastera na osnove tematicheskogo predstavleniya], Computational Linguistics and Intelligent Technologies: Proc. The International Conference Dialog'2009, [Komp'yuternaya lingvistika i intellektual'nyye tekhnologii: trudy Mezhdunarodnoy konferentsii Dialog'2009] Vol. 8 (15). c. 299–305.
16. *McKeown K. R.* (1985), Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text. Cambridge University Press.
17. *Saggion, H. A.* (2009), Classification algorithm for predicting the structure of summaries. Proc. The 2009 Workshop on Language Generation and Summarisation, ACL-IJCNLPЬ 2009. — Suntec, P. 31–38.
18. *Sheremetyeva S.* (2005), Embedding MT for Generation in English from a Multilingual Sheremetyeva S. 2005. Embedding MT Generation in English from a Multilingual Interface. Proceedings of the workshop on Patent Translation in conjunction with MT Summit X, Phuket, Thailand, September, 12–16.
19. *Sheremetyeva S.* (2012), Automatic Extraction of Linguistic Resources in Multiple Languages. In Proceedings of NLPCS 2012, 9th International Workshop on Natural Language Processing and Cognitive Science in conjunction with ICEIS 2012, Wroclaw, Poland.
20. *Thiago A., H. Rino and M. Nunes.* (2003), GistSumm: a summarization tool based on a new extractive method. In Proceedings of the 6th international conference on Computational processing of the Portuguese language, pages 210–218.
21. *Underwood N. L. and Jongejan B.* (2001), Translatability Checker: A Tool to Help Decide Whether to Use MT. Proceedings of MT Summit VIII, Santiago de Compostela, Spain.
22. *Wan S., Robert Dale Mark Dras, and C´ecile Paris.* (2005), Towards statistical paraphrase generation: preliminary evaluations of grammaticality. Proc. The 3rd International Workshop on Paraphrasing (IWP2005), Jeju Island, South Korea, pp. 88–95.
23. *Wan S., Robert Dale Mark Dras, and C´ecile Paris.*(2008), Seed and Grow: Augmenting Statistically Generated Summary Sentences using Schematic Word Patterns. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 543–552, Honolulu, October 2008.
24. *Wubben S., A. van den Bosch and E. Krahmer.* (2012), Sentence Simplification by Monolingual Machine Translation. In Proceedings of the 50th Annual Meeting of the Association for Computational linguistics. Jeju, Korea, July.