Thesaurus tool for analysing the semantic compatibility of educational texts

G R Rybakova¹⁴, A Yu Andreeva², I V Krotova⁴, T L Kamoza¹ and O V Chashchina³

¹ Siberian Federal University, 79 Svobodny Ave., Krasnoyarsk, 660041, Russia
² Polzunov Altai State Technical University, 46 Lenin Ave., Barnaul, 656038, Russia
³ Municipal autonomous educational institution secondary school No. 1 named after V. I. Surikov, 7B Krasnodarskaya St., Krasnoyarsk, 660005, Russia

⁴ E-mail: rbkv@yandex.ru

Abstract. The developed software is designed to analyze the semantic compatibility of educational text materials. This need is due to the existing negative trends noted by scientists (didactics, psychologists, neuroscientists) in the mechanism of information perception in the learning process of the modern generation of students. The way to level this tendency in the educational process is to take into account the unconscious patterns of information perception, associated with a gradual increase in its complexity. Semantic compatibility in the developed program is assessed by comparing the thesauri of educational materials, which allows in the future to influence the required degree of increase in their novelty (and, as a consequence, complexity). The work algorithm is formed taking into account the specific variable features of the Russian language. The proposed analytical tool is intended for didactic use, but due to the flexibility of the action algorithms provided by the program, it allows the use of the software product for the analysis of texts for other purposes, for which compatibility is of practical importance.

The need to improve the efficiency of assimilation of information in the educational process is due to the observed trends in changes in the information space, educational technologies, in the work of the brain of modern generations of students [1-7]. Semantic compatibility of educational texts becomes one of the factors contributing to a more complete perception of new information [8-12], assimilation of material due to a reasonable balance between new and already familiar [13-15], attracting an unconscious component to the educational process [3-5; 16-23].

Thesaurus measure allows one to quantitatively express the semantic level of communication between educational texts [15; 24-28], which makes it possible to form complexes of educational texts with a gradual increase in difficulty levels, not just without prejudice to the quality of mastering the material, but with the expected increase in efficiency due to the scientifically confirmed specificity of the unconscious component of perception. Thus, the purpose of the developed software for the educational and methodological process is to create a tool for analysing the semantic textual compatibility of educational materials for their subsequent possible and necessary correction.

Building a thesaurus automatically is not an easy task. Since the machine does not know how to understand the text the way a human does, this task requires breaking it down into several stages [29-
32):
1. text preprocessing;
2. selection of candidates;
3. calculating the weights of candidates;
4. calculating the range of weights that will be included in the thesaurus.

1. Text preprocessing is a process that prepares information for a form suitable for analysis. At this stage, normalization is necessarily carried out - reducing the word to its initial form. For these purposes, it is customary to use two approaches: stemming - reduction by cutting off endings, prefixes (as a result of this, the base remains - the root of the word) and lemmatization - when they come to the initial form in the process of morphological analysis. This method is the most logical and reliable, since during stemming you can spoil the word.

Based on the rules of the Russian language, we will understand the dictionary form as: for nouns - words in the singular of the nominative case, for adjectives with similar parameters (singular, nominative), a gender requirement (masculine) is added; for verbs, the infinitive form is applicable.

Carrying out lemmatization of words is based on the use of morphological analysers, which usually consist of a tagger and a lemmatizer. Tagger conducts part-of-speech mark-up of words to determine grammatical features. At this stage, the task is realized in the course of automatic processing of individual words of a text material to identify a part of speech and a set of grammatical characteristics inherent in a word that allow assigning an appropriate set of tags to it, after which the entire resulting array is transferred to the lemmatizer to reduce the word to its initial form.

The specificity of the Russian language imposes a number of restrictions on the use of morphological analysers due to the complex structural organization of the language associated with its prevailing inflectional nature. This property is due not only to analytical complexity, but to a greater extent inherent in it a high level of flexibility, in which such languages are characterized by variability of words (gender, case, number in the endings) due to the multidimensionality of meanings fixed in formants (inflections). In the presence of such multidimensionality, the definition of the lemma in the case of multivariance of word forms objectively complicates the normalization process.

Among the analysers of the morphology of words for the Russian language, the most demanded is the analyser designed in the Python language - Pymorphy2. For work, it uses the OpenCorpora dictionary in the form of a file, in which words are combined into tokens according to the principle: the first in the list is the normal form, then all forms of the word for this lexeme are shown with grammatical information for each of them.

The morphological analyser is able not only to parse a word, but also to decline. When parsing a word, the analyser looks for it in the dictionary and, after finding it, returns the grammatical information assigned to it. When declining, a search is also performed in the dictionary, a lexeme is determined, from which the analyser returns the desired form.

The OpenCorpora dictionary for the Russian language contains about 400 thousand lexical items and 5 million individual words. However, it is impossible to cover all the words of the Russian language, and if the analyzer comes across a word that is not in the dictionary, the matter is entrusted to “predictors” - the rules for parsing unknown words.

Pymorphy2 predicts single-letter words in uppercase as initials, for them the parsing options "name" and "patronymic" are returned, for all genders, cases and numbers. However, this operation is not always performed correctly, which is why it was decided to select all abbreviations and named entities before normalization, so as not to spoil them at the stage of normalization.

Maru is a morphological analyser written in Python. Uses a lemmatizer from Pymorphy2 and a tagger any of the following:
1. Pymorphy2;
2. Linear;
3. CRF;
4. RNN (figure 1).
When using the Pymorphy2 tagger, the result will be the same as that of the Pymorphy2 morphological analyser itself. RNN is more accurate out of all four, but the operating speed is the slowest. CRF is most optimal, it is not too slow, and the quality of work is good enough.

If we compare the work of this morphological analyser with Pymorphy2, while the tagger is using CRF, then Maru will work slower, but better.

After the studies of these morphological analysers, it was concluded that the quality of lemmatization depends not only on the completeness of the dictionary, but the choice of the tagger plays an important role. If the grammatical features of a word are determined incorrectly, then there will be errors during normalization.

During text preprocessing, it is advisable to select named entities and work with them personally, since there is a high probability of this data corruption (figure 2).

The task of extracting named entities was first posed at the sixth MessageUnderstanding conference in 1995 and included recognition of object names for people and organizations, geographic names, temporary expressions, and some types of numeric expressions. In the linguistic sense, this is the designation of a category (object, phenomenon) in the form of a word or a combination of words. Named entities include:

- names of personalia;
- names of organizations;
- geographical names;
- brands of goods;
Natasha package - solves the main tasks of natural language processing for the Russian language: tokenization, sentence segmentation, morphology tagging, lemmatization, phrase normalization, parsing, NER tagging, fact extraction. The package uses grammars and dictionaries for the Yargy parser.

Natasha extracts the following entities:
- individuals: Ivanov Ivan Ivanovich, Ivan Ivanov, Ivan Petrovich, Vanya;
- organizations: PJSC Gazprom, IP Ivanov Ivanovich, IP Ivan Ivanov, Cosmonaut club, Hermitage, State Duma of the Russian Federation;
- events: the festival “The Ark of Salvation”, the show “The Fifth Empire”;
- geo-objects: Moscow, Leningrad Region, Russian Federation, North Caucasian Federal District;
- time objects: May 21, 1996, 05/21/1996, May 21, today, at the end of the year;
- monetary units: 200 rubles, 1 billion dollars;

However, not all of them are needed to solve our problem. Monetary units can be safely discarded. When extracting, data is written to structures in a normalized form.

The first stage - text preprocessing - is carried out in order to filter out unnecessary data and bring the rest to a form suitable for identifying candidates for a thesaurus.

The algorithm for highlighting words is as follows:
1. highlighting sentences;
2. clearing stop words;
3. selection of named entities;
4. highlighting the abbreviation;
5. cleaning from “garbage”;
6. conversion to lower case;
7. normalize;

1.1. The selection of sentences as the first step of the word extraction algorithm involves breaking the text into sentences by calculating the weights of future candidates. Partition options:
1. re.split([.?!…]);
2. segtok.split_single;
3. mosestokenizerltk.sent_tokenize;
4. deepavlov/rusenttokenize;
5. razdel.sentenize.

The first option was discarded, since in this case many mistakes are made. Second to fourth are not only slow, but also inaccurate.

![Figure 3. Comparison of tokenizers](image_url)
The automatic tokenizer does not always manage to unambiguously divide the text into separately significant units; the text may contain abbreviations or abbreviated words that can be shared as several separate units during processing. The best option is the Razdel library, which is based on the rules of the Russian language and provides for such cases (figure 3).

1.2. The second step for highlighting words is to clear the stop words. Stop words include pronouns, prepositions and other words that do not carry a semantic load. The Stopwords library from the NLTK.corpus package, which contains many lists of stop words, including for the Russian language, helped to clear them of them. Due to the small volume, I had to add to the list manually. At the moment, 701 words are recorded in the list.

1.3. The extraction of named entities (the third step of the word extraction algorithm) is performed using the Natasha package in order not to spoil them when processing text later. Since the names of persons, or the names of geographic places, may suffer during normalization.

1.4. The acronym highlighting, the fourth step of the algorithm, is implemented using regular expressions (a formal search language) and the re module. At this stage, manipulations are carried out using metacharacters with respect to text substrings. The search rule is specified by a string of characters and metacharacters, which serves as a pattern:

```python
re.findall(r'^[a-zA-Z]+[a-zA-Z]*[^a-zA-Z]*', text)  # selects words with more than two letters
```

1.5. The step of the algorithm associated with cleaning up “garbage” is implemented using the re module, thanks to which numbers, punctuation marks and other symbols are eliminated, and then those words that are longer than two are selected:

```python
re.compile(r'^[^a-zA-Z]*')  # selects only words, eliminating numbers and various characters.
```

1.6. Conversion to lowercase (the sixth step of the word extraction algorithm) using the lower function is carried out so that no corresponding difficulties arise during normalization. Words that are in OpenCorpora are written in lowercase.

1.7. Normalization (step 7) is necessary so that the same word does not get into the thesaurus, but in different forms. On the one hand, as a result, the weight will be calculated incorrectly, on the other hand, a significant word may not be included in the thesaurus at all.

Although normalization belongs to the stage of text preprocessing, during the implementation of the algorithm it smoothly flowed into the second stage - the selection of candidates for the thesaurus. This is discussed in more detail below.

2. Selecting candidates for the thesaurus. At this stage, the remaining text is divided into words and phrases. When selecting candidates, a simple division by one word will not work, since after it the “differential equation” will be divided into “differential” and “equation”, which is not very good. That is why it is necessary to provide for the correct division.

After normalization, an attempt was made to split each sentence into two words, and then search in the entire text for a pair that was repeated at least once. However, this method of highlighting words and phrases takes a long time.

To solve this problem, the rutermextract library was found, which not only broke sentences into words and phrases, but also normalized them. She puts adjectives that stood with nouns in the desired case, and for the rest returns the normal form. For normalization, the library uses the Pymorphy2 morphological analyser. There is no documentation in rutermextract, but there are enough comments in Russian in the code, which makes it easy to understand how the library works.

3. The third stage is the calculation of the weights of the thesaurus candidates. It is carried out by calculation based on statistical dimension, which determines the frequency (significance) of each of the words within the context of the set of training materials (TF×IDF metric). The TF frequency (the significance of the word t in the document) is calculated by the formula (1) by the ratio of the number of its occurrences in the text to the total number of words in the document:
\[
\text{tf} \left( t, d \right) = \frac{n_t}{\sum_k n_k}
\]

where \( n_t \) is the number of occurrences of the word in the document; 
\( \sum_k n_k \) – the sum of occurrences of all words in the document.

The formula for calculating IDF (2), the reciprocal of the frequency of occurrence of a word in a set of documents, may not take into account the base of the logarithm, since there will be no effect on the ratio of weights when changing by a certain factor:

\[
\text{idf} \left( t, D \right) = \log \frac{|D|}{|\{d_i \in D | t \in d_i\}|}
\]

where \( |D| \) – the number of documents in the collection; 
\( |\{d_i \in D | t \in d_i\}| \) – the number of documents from collection D, in which the word \( t \) occurs.

The formula for determining the TF-IDF measure is the operation of multiplying TF by IDF (3):

\[
\text{tf} - \text{idf} \left( t, d, D \right) = \text{tf} \left( t, d \right) \times \text{idf} \left( t, D \right).
\]

To calculate TF-IDF, there is a TfidfVectorizer class from Sklearn, the input of which is a string, and the output is a list of words and their values. Thus, the class processes the text on its own: divides into words, converts to lowercase and calculates the metric. Since we need to calculate the weight for already selected words and phrases, it was decided to implement the algorithm ourselves.

4. The final, fourth stage of building a thesaurus is calculating the range of weights that will be included in the thesaurus. After the candidates are selected, you need to select the more significant ones, that is, those that will be included in the thesaurus. According to Zipf's law, ordering a set of words according to the frequency of their use, there is an inverse relationship between the rank of a word in this ordering (ordinal number \( n \)) and the frequency of using this word [33-36]. The most popular words will appear in most sentences. They are often functional words with no meaning. Therefore, to measure the weight of words, it is better to take the TF-IDF metric in order to reduce the weight of words that are often used in the text and do not carry a semantic load.

After that, you need to select a range from which words will be selected in the thesaurus. The quality of the future thesaurus depends on it, because if you set a large width, then words that are auxiliary will fall into the range; if you set a narrow range, you can lose important terms. Therefore, to determine the width of the range is determined heuristically. After the research carried out, it was decided to take the largest weight as the beginning of the range, since the preprocessing of the text is carried out quite efficiently and words without semantic load do not have much weight. The last thing left to figure out is the end of the range.

In computational linguistics, the empirical law of G. S. Hips connects the volume of all words of a text with the volume of especially important words that are included in this text. According to him, these values are related by the formula (4):

\[
v(n) = \alpha n^\beta,
\]

where \( v \) is the volume of all words, composed of a text that consists of \( n \) particularly important ones, 
\( \alpha \) and \( \beta \) are empirically determined parameters For European languages, \( \alpha \) takes a value from 10 to 100, and \( \beta \) – from 0.4 to 0.6.

The sought value is \( n \), which will be the final boundary of the thesaurus range. After it is found, from the candidates are selected those that lie in the range \([0, n]\).

The main purpose of the obtained software tool for analysing textual educational materials is to analyse the compatibility of information in educational materials of different topics, sections, disciplines, scientific branches. Therefore, the thesaurus formed at the four previous stages is the basis for the second, most significant for achieving the set goal, part of machine processing. After the thesaurus of educational materials are formed, they are analysed for compatibility. To calculate the compatibility of thesaurus, the following formula was used (5):
\[ S_t = \frac{N_o/N_{T1} + N_o/N_{T2}}{2} \times 100\%, \] (5)

where \( S_t \) – compatibility of two compared thesaurus, expressed in \%; \( N_o \) – the number of common elements for all thesaurus; \( N_{T1} \) – the number of elements of the first of the compared thesaurus; \( N_{T2} \) – the number of elements of the second of the compared thesaurus.

The resulting compatibility values can be considered optimal if the resulting values are close to 85%. In this case, according to studies [13, 15, 37], we can talk about the increment of the thesaurus within the necessary balance of new and already familiar information. In this case, it makes sense to analyze consistently studied topics, sections where the introduction of new concepts, definitions should be about 15%. At the same time, as noted by researchers [13, 15, 37], a decrease in the share of new concepts just as does not contribute to the preservation of attention, learning motivation and perception of information, as well as a noticeable excess of this share.

The developed software product is a didactic-analytical tool suitable for monitoring the degree of increase in the complexity of information [13, 15, 37] subject to assimilation, and its subsequent adjustment and optimization of the parameters of educational texts. This, in turn, makes it possible to reduce the influence of existing non-positive trends in the context of digital transformation of society, the consciousness of students [6-7], and continuous education [16].

When testing the program on a set of educational texts of a number of related disciplines, the developed software showed the adequacy of work to achieve the set goals. The flexibility of its algorithms also indicates the possibility of a more universal application of the program with respect to texts of any purpose, where semantic compatibility matters.

References
[1] Anokhin KV and Velichkovsky BM 2011 Russian Foundation for Basic Research Journal 2-3(70-71) 67-77
[2] Chernigovskaya TV 2010 On the Ways to a New School 1 11-3
[3] Chernigovskaya TV et al. 2016 Psychophysiological and Neuralinguistic Aspects of the Process of Recognizing Verbal and Non-verbal Communication Patterns (St. Petersburg: VVM)
[4] Damasio A 2018 This is How the Self Begins. The Brain and the Emergence of Consciousness (Moscow: Career-Press)
[5] Dean S 2018 Consciousness and the Brain. How the Brain Encodes Thoughts (Moscow: Career-Press)
[6] Feldstein DI 2013 Russian Psychological Journal 10(2) 7-31
[7] Klochkov V P, Rybakova G R and Malkova T V 2020 Modern Scientist 1 109-15
[8] Klochkov V P et al. 2019 Amazonia Investiga 8(23) 506-16
[9] Rybakova G R, Krotova I V and Kamoza T L 2015 Humanitarian Vector 1(41) 24-34
[10] Rybakova G R, Andreeva A Yu and Katkov A S 2017 Program for modeling the compatibility of educational information “Thesaurus” Certificate of registration of a computer program RU 2017662610
[11] Rybakova G R, Andreeva A Yu, Krotova I V and Kamoza T L 2020 Software for Sense Compatibility Analysis of Educational Texts European Proc. of Social and Behavioural Sci. EpSBS 90 800-8
[12] Vasileva N O, Rybakova G R and Shevtsova L N 2020 Professionally Oriented Terminology Databases as an Element of the Educational Process European Proc. of Social and Behavioural Sci. EpSBS 90 1515-25
[13] Metcalfe J. 2017 Learning from errors Annu. Rev. Psychol. 68 465–89
[14] Mai F, Galke L, Brunsch D and Scherp A 2017 K-CAP 2017: Proc. of the Knowledge Capture Conf. Decembe 20 1-4 Retrieved from: https://doi.org/10.1145/3148011.3148039
[15] Wilson R C et al. 2019 Nat Commun 10 4646 Retrieved from: https://doi.org/10.1038/s41467-019-12552-4
[16] Verbitskaya O 2019 Bulletin of the South Ural State University. Ser. Education. Educational Sciences 11(3) 6–20
[17] Klemantovich I P, Levanova E A and Stepanov V G 2016 Pedagogy and Psychology of Education 2 8-17
[18] Klochkov V P, Kamoza T L, Krotova I V and Donchenko N A 2008 Humanitarian Vector 1 41-5
[19] Kryukova A P, Agafonov A Yu, Kozlov D D and Shilov Yu E 2017 Bulletin of Kemerovo State University 2 158-63
[20] Morozov M I and Spiridonov V F 2019 Vestnik of Saint Petersburg University. Psychology 9(3) 280–94 Retrieved from: https://doi.org/10.21638/spbu16.2019.305
[21] Pervushina N A and Osetrin K E 2017 Pedagogical Review 2(16) 198-208
[22] Filippova M G 2006 Experimental Psychology of Cognition: Cognitive Logic of the Conscious and the Unconscious (St. Petersburg: St. Petersburg University Press)
[23] Schenk R 2012 Cognitive Studies of Language 5 289-90
[24] Groot F, Huettig F and Olivers C 2016 Journal of Experimental Psychology: Human Perception and Performance 42(2) 180–96
[25] Lagutina N S, Lagutina KV, Mamedov E I, and Paramonov I V 2016 Model. Anal. Inf. Sist. 23(6) 826–40
[26] LeCun Y, Bengio Y and Hinton G 2015 Nature 521 436-44
[27] Mai F, Galke L, Brunsch D and Scherp A 2018 JCDL '18: Proc. of the 18th ACM/IEEE on Joint Conf. on Digital Libraries 169–78 Retrieved from: https://doi.org/10.1145/3197026.3197039
[28] Martinez-González M M and Alvite-Díez M I 2016 Computer Standards & Interfaces 65(July) 79-91
[29] Luyi S, Xin Y, Xiaojin W, Haitao W, Xinyu C and Liangliang L 2019 Overview and development of Chinese and English thesauri WI ’19 Companion: IEEE/WIC/ACM Int. Conf. on Web Intelligence - Companion October 165–70 Retrieved from: https://doi.org/10.1145/3358695.3360942
[30] Tsarev R Y et al. 2019 Cybernetics and Automation Control Theory Methods in Intelligent Algorithms CSOC 2019 986 Retrieved from: https://doi.org/10.1007/978-3-030-19813-8_34
[31] Liebeskind C, Dagan I and Schler J 2019 Applied Artificial Intelligence 33(4) 483-96 (6) https://doi.org/10.1080/08839514.2019.1583447
[32] Moreno-Sánchez I, Font-Clos F and Corral Á 2016 PLoS ONE 11(1) e0147073. Retrieved from: https://doi.org/10.1371/journal.pone.0147073
[33] Piantadosi S T 2014 Psychonomic Bulletin & Review 21(5) 1112-30
[34] Qiu J, Zhao R, Yang S and Dong K 2017 Informetrics. Springer, Singapore Retrieved from: DOI https://doi.org/10.1007/978-981-10-4032-0_5
[35] Yatsko V A 2015 Automatic Documentation and Mathematical Linguistics 49 83-8 Retrieved from: https://doi.org/10.3103/S0005105515030048
[36] Shenhav A et al. 2017 Annu. Rev. Neurosci. 40 99-124