

# Towards semantic search for mathematical notation

Agnieszka Bier

Institute of Mathematics

Silesian University of Technology

23 Kaszubska Str.

44-100 Gliwice, Poland

Email: agnieszka.bier@polsl.pl

Zdzisław Sroczyński

Institute of Mathematics

Silesian University of Technology

23 Kaszubska Str.

44-100 Gliwice, Poland

Email: zdzislaw.sroczyński@polsl.pl

**Abstract**—The paper concerns the design and implementation of a search engine for mathematical expressions given by the user in a convenient form of natural language or visual queries. Proper presentation and transcription of the mathematical notation is substantial for further processing and the adequate choice of the word distance measure for string comparison is an important issue as well. Within this project a complete solution for acquiring and processing the mathematical query and a searching algorithm is elaborated. We present results of exemplary search queries obtained for different types of input data format with application of two different word distance measures and discuss briefly the observed properties.

## I. INTRODUCTION

THE mathematical notation is an example of a visual language, which uses images for communication. A visual language consists of a set of planar diagrams representing expressions which are defined to be correct, and in which the syntax and semantics are determined by the mutual geometric relations between particular symbols.

The mathematical notation intended for machine processing may be transcribed in many different formats. The most popular ones are  $\LaTeX$ , a dedicated document preparation system widely used for publication of scientific content, and MathML dedicated to Web applications. Both of these formats have their origins in text description languages,  $\TeX$  and HTML/XML, respectively, and are designed to be open. Other solutions are used mainly in Desktop Publishing (DTP) and usually allow exporting to and importing from the above mentioned universal formats. We also mention a special format for mathematical Braille dialect, used by visually impaired people.

Independently of the chosen document description language, automation of mathematical notation processing requires transcription of the common planar visual language notation into a sequential notation used in programming languages. Hence all structural information on the mathematical formula, encoded in the geometric mutual relations between symbols, must be transcribed to the notation, in which the semantics of the given expression is incorporated solely in the sequencing of symbols possibly grouped by brackets (the significant dissimilarity of the Reverse Polish Notation to the natural mathematical notation excludes its application in most of the areas, especially for educational purposes). Such approach enables preparation of documents containing arbitrary complex mathematical expressions, while on the other hand it provides

poor capabilities for searching or indexing of mathematical content.

The goal of the presented work was to develop a novel tool for searching for mathematical expressions based on queries containing parts of original or similar formulae, which are preferably given in a natural language.

## II. PROCESSING MATHEMATICAL NOTATION FOR SEMANTIC SEARCH

The problem of expressing the mathematical symbolic notation in the natural language has drawn interest of many researchers in different aspects within the scope of didactics of mathematics. A particular interest in this issue is observed in terms of assistance for the disabled people. A thorough survey on the English transcription of mathematical content can be found in [1], while some considerations on the ambiguity of certain notation rules are elaborated in [2], [3], [4], [5]. Various examples of algorithms implementing verbalization of mathematical content can be found in [6], [7], [8], [9], [10] and an algorithm designed to meet the needs of visually impaired is presented in [11]. Other results concern the problem of searching the mathematical expressions (see [12], [13], [14], [15], [16]) and some indexing algorithms for efficient search are elaborated in [17], [18], [19], [20], [21]. The automatization of content categorization is discussed in [22], methods for correctness validation are presented in [23], and certain search algorithms by means of natural language queries are elaborated in [24], [25], [26] and [27].

The verbalization of mathematics provided by math-to-speech engine used in our experiments allows the use of different dialects and national versions. There are English, Polish simplified/natural and full transcriptions available at the moment. The main difference between these versions is the level of details and abbreviations introduced to shorten the output and make it sound more or less similar to the actual verbalization done by human readers.

To build the query in an artificial description language with the use of complex notation and the set of non-familiar commands the user must have extensive technical experience. This is very unlikely for the majority of students at different levels of education. Moreover, translating the query into the special language distracts the user's attention from the actual search object.

There is also the additional challenge – to construct the query with the use of some characteristics of the searched mathematical expression, for example the presence of the particular key symbol as root or integral sign. This is important because the user very seldom remembers exactly the whole equation, rather reminds only some fragments, the general layout or is able to prompt the field of mathematics.

Summarizing, the important issue while searching for the mathematical expression in the database containing the collection of equations is the construction of the query. It can be done with the use of common document description languages, but it requires complex lexical knowledge and complicates queries based on the fragmentary information and simplifications due to imperfections of human memory.

The verbalization of the mathematical notation stored in the database can help to solve problems mentioned above. There is a possibility to compare the spoken version of the equation with the query given in natural language with the use of the methods, tools and measures for approximate string matching. This way the accuracy of the semantic search for mathematical notation should significantly increase.

Given a query expression, the search algorithm follows the steps:

- 1) Converting the query expression entered in the Formula Editor into one of three formats:  $\LaTeX$ , natural English, natural Polish
- 2) Comparison of the transcribed query with the existing database records by means of one of the text similarity measures: normalized Levenstein distance, and (weighted) cosine similarity measure.
- 3) Visual presentation of the list of 20 best matches (according to the chosen comparison measure results)

The Levenshtein distance  $D_{Le}(a, b)$  is the measure of difference between text strings  $a$  and  $b$  of possibly different length, calculated as the minimum number of single-char edit operations, required to change one string ( $a$ ) into another ( $b$ ) [28], where operation categories are: insertions, deletions and replacements weighed equally 1. The normalized Levenshtein distance  $D_{LeN}(a, b)$  additionally places the resulting values in the range 0 – 100%

$$D_{LeN}(a, b) = 100 \times \frac{D_{Le}}{\max(|a|, |b|)}$$

where  $|s|$  is number of characters in string  $s$ .

Another useful text comparison measure is the (weighted) cosine similarity measure. Two strings are first transformed into numerical vectors  $v = [v_1, \dots, v_n]$  and  $w = [w_1, \dots, w_n]$  with coordinates being the counts of instances of a given key word in the string, and then the cosine of the angle between  $v$  and  $w$  is calculated by means of the standard dot product of the two vectors:

$$\cos \angle(v, w) = \frac{v \circ w}{|v| \cdot |w|} = \frac{\sum_{i=1}^n v_i \cdot w_i}{\sqrt{\sum_{i=1}^n v_i^2} \cdot \sqrt{\sum_{i=1}^n w_i^2}}$$

Values close to 1 are interpreted for high similarity of the vectors, while values close to 0 indicate big differences in  $v$  and  $w$ . Additionally, by introducing weights in the cosine similarity measure, one can choose symbols that should have stronger or weaker impact on the comparison. For instance, since brackets are typically used only for ordering the expression and do not introduce their own meaning, one should pay less attention to them while comparing formulae. In our examples the following weights have been assigned to certain basic types of mathematical expressions: roots 30, fractions: 30; integral: 30; sums: 30; components: 15; products: 30; factors: 15; limits: 30; parentheses and brackets: 5; powers, including squares and cubes: 20. These weights have been chosen experimentally as ones appearing the most suitable for effective comparison.

In the case of comparison based on the verbalization (word transcription) of the formulae, an initial step of conversion to the spoken language is performed and the transcripts are stored in a database. In the conversion process the macrodefinitions formulated in an interpretable script language Lua are used. A model is loaded from the database to the inner engine, which stores the tree of objects corresponding to each element of the processed formula. To increase the efficiency of the comparison all database records are pre-processed and saved in an additional data structure of type in-memory-table. As the processing time of a formula consisting of tens of symbols is ca 50–100ms for typical PC unit, the initial pre-processing step significantly increases the performance of the presented algorithm.

### III. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented and tested for the quality of search, interpreted as the similarity of the original formula to the obtained results. The performance measure of the search algorithm was defined as the position of the original formula on the list of search results. The experimental queries were formulated in all accepted data formats:

- 1) the inner data format of the Equation wizard (“Edytor wzorów”) application, based on  $\LaTeX$  language,
- 2) the natural English transcription/verbalization,
- 3) the natural Polish transcription/verbalization.

With respect to the details of the content and possible application areas the following three categories of test queries were distinguished:

- (C1) the query and original expressions differ in single symbols (correction),
- (C2) the query was a part of the original expression (autocompletion),
- (C3) the query expression contains few symbols, specific to the original expression (search).

Each test query was introduced to the search engine in two ways: by means of a visual editor and by entering the respective natural language key words. The performance of our search algorithm was then compared for the different

TABLE I  
TOTAL SUCCESS RATE OF SEARCH WITH RESPECT TO FORMATS OF COMPARED DATA, QUERY INPUT (VISUAL / KEY WORDS), AND THE MEASURE USED FOR STRING COMPARISON.

Data format	Measure	Visual query	Key words query
Inner	Levenstein	66.6%	0%
	Cosine	66.6%	17%
English	Levenstein	83.3%	100%
	Cosine	100%	50%
Polish	Levenstein	100%	83.3%
	Cosine	100%	33.3%

data formats, within each category of queries and the input modes. The test consisted of two parts. Firstly, the queries were executed on a general database of mathematical expressions, containing various formulae from different branches of mathematics. Some of the database records were totally irrelevant to the queries. We present the obtained results in Table II, column GD.

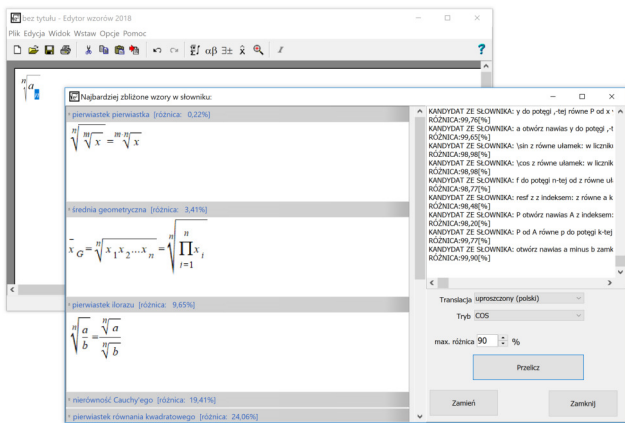


Fig. 1. The user interface of the Equation wizard („Edytor wzorów”) test application with the results of the visual query.

For the second part of the test a specialized database was used in order to reveal possible weaknesses of the search algorithm or text similarity measures used for comparison. The test database consisted of 20 different formulae all containing quadratic subexpressions and having elements identical with the original formula. The column SD in Table II presents the search results for three different queries referring to the original formula  $(a + b)^2 = a^2 + 2ab + b^2$  in different settings.

In order to compare the performance of the search algorithm for different settings, the total success rate has been calculated. The total success rate is defined to be the percentage of correctly found expressions, where correctly found means that the desired expression was among the first five entries on the list of search results. A summary of these rates is presented in Table I. The main observations are the following. The queries in lingual data formats provide the required expressions at average rate of more than 81%, while the success rate of

queries processed from inner format is twice lower (around 37%). A less significant difference is observed when comparing the similarity measures used by the search algorithm - the Levenstein measure success rate is around 72%, while for cosine measure it is around 61%. However, if one looks into positioning of the queried formula, the advantage of Levenstein measure becomes more evident. Finally, the observed success rate of visual queries is significantly higher than for key word queries – nearly a half (47%) of key word queries provided desired outcome, while 86% of the visual queries appeared successful.

A few further results, presenting the overall performance issues of the proposed algorithm on a general database are given in Table III.

IV. CONCLUSIONS

The results of the experiments with the developed interactive application give the basis for the research of an efficient mathematical search engine. Both the search with the description languages as  $\LaTeX$  as well as usage of verbalized text fragments was possible and accurate, especially while processing relatively not much different equations. Therefore, the proposed method is suitable for automatic or semi-automatic, supervised correction of documents containing complex mathematical notation.

On the other hand, the search for the given equation on a partial information basis, i.e. the excerpt of the formula or unordered sequence of symbols or keywords, has been also successful in many cases. Thanks to developed solution such an approximate matching is convenient and does not require any extra technical knowledge from the user, since the system accepts queries in natural language. The introduction of the advanced translator for the verbalization of the mathematical notation, taking into account the complexity of the sub-expressions, dialect and national languages helped to increase the quality of text comparisons and thus the efficiency of the search.

The performed tests revealed also some general tendencies in the results obtained for different kinds of queries when compared by means of different similarity measures. The Levenstein measure seems more suitable for queries that differ slightly from the origin and therefore it could be applied for corrections. On the other hand, the cosine measure appeared to provide better matching for shorter queries, being parts of the original formula, and hence its area of application would preferably be searching or autocompletion.

One of the main directions for future experiments in this area could be the introduction of more flexible queries, possibly with the notation for logical operators as AND, OR and NOT, which could increase the accuracy of the search.

REFERENCES

[1] L. A. Chang, *Handbook for Spoken Mathematics (Larry's Speakeasy)*. Lawrence Livermore Laboratory, The Regents of the University of California, 1983.  
 [2] R. Fateman, "Handwriting + speech for computer entry of mathematics," *Style*, Benjamin L. Kovitz, Manning Publications Company, 2004.

TABLE II

SEARCH RESULTS FOR DIFFERENT PARTS OF FORMULA IN A GENERAL DATABASE (GD) AND SPECIALIZED DATABASE (SD) WITH RESPECT TO FORMATS OF COMPARED DATA (INNER – LATEX, ENG – NATURAL ENGLISH, POL – NATURAL POLISH), QUERY INPUT (VISUAL / KEY WORDS), AND THE MEASURE USED FOR STRING COMPARISON (L - LEV, COS - COSINE). LAST COLUMN PRESENTS THE POSITION OF THE INTENDED ORIGINAL FORMULA  $(a + b)^2 = a^2 + ab + b^2$  ON THE LIST OF RESULTS (\* – FORMULA OUTSIDE THE LIST OF 20 BEST MATCHES).

Query	Data format	Input Mode	GD		SD	
			Pos. L	Pos. COS	Pos. L	Pos. COS
Category: C1 (correction)						
$(a + b)^2 = a^2 + ab + b^2$	inner	visual	1	1	1	1
"square of sum of a and b equals a squared plus ab plus b squared"	inner	key words	*	*	9	7
"kwadrat sumy a i b równa się a kwadrat plus ab plus b kwadrat"	inner	key words	*	*	12	7
$(a + b)^2 = a^2 + ab + b^2$	Eng	visual	1	1	1	1
"square of sum of a and b equals a squared plus ab plus b squared"	Eng	key words	1	1	1	5
$(a + b)^2 = a^2 + ab + b^2$	Pol	visual	1	1	1	1
"kwadrat sumy a i b równa się a kwadrat plus ab plus b kwadrat"	Pol	key words	1	9	1	4
Category: C2 (autocompletion)						
$(a + b)^2$	inner	visual	14	7	4	2
square of sum of a and b	inner	key words	*	*	12	9
kwadrat sumy a i b	inner	key words	*	*	12	9
$(a + b)^2$	Eng	visual	2	3	1	2
square of sum of a and b	Eng	key words	9	*	4	9
$(a + b)^2$	Pol	visual	2	3	1	2
kwadrat sumy a i b	Pol	key words	7	8	5	8
Category: C3 (search)						
$()^2$	inner	visual	*	9	5	5
squared	inner	key words	*	*	12	7
do kwadratu	inner	key words	*	3	12	7
$()^2$	Eng	visual	8	2	1	3
squared	Eng	key words	8	2	3	6
$()^2$	Pol	visual	5	2	1	3
do kwadratu	Pol	key words	5	2	3	6

- [3] —, "How can we speak math?" Computer Science Division, EECS Department, University of California at Berkeley, Tech. Rep., 2013.
- [4] J. Cuartero-Olivera, G. Hunter, and A. Pérez-Navarro, "Reading and writing mathematical notation in e-learning environments," *eLearn Center Research Paper Series*, no. 4, pp. 11–20, 2012.
- [5] Z. Sroczyński, "Priority levels and heuristic rules in the structural recognition of mathematical formulae," *Theoretical and Applied Informatics*, vol. 22, no. 4, p. 273, 2010.
- [6] A. Bier and Z. Sroczyński, "Adaptive math-to-speech interface," in *Proceedings of the Multimedia, Interaction, Design and Innovation*, ser. MIDI '15. New York, NY, USA: ACM, 2015. doi: 10.1145/2814464.2814471. ISBN 978-1-4503-3601-7 pp. 7:1–7:9. [Online]. Available: <http://doi.acm.org/10.1145/2814464.2814471>
- [7] D. Attanayake, J. Denholm-Price, G. Hunter, E. Pfluegel, and A. Wigmore, "Intelligent assistive interfaces for editing mathematics." in *Intelligent Environments (Workshops)*, 2012, pp. 286–297.
- [8] D. Attanayake, G. Hunter, J. Denholm-Price, and E. Pfluegel, "Novel multi-modal tools to enhance disabled and distance learners' experience of mathematics," *ICTer*, vol. 6, no. 1, 2013.
- [9] T. Sancho-Vinuesa, C. Córcoles, M. Huertas, A. Pérez-Navarro, D. Marquès, R. Eixarch, and J. Villalonga, "Automatic verbalization of mathematical formulae for web-based learning resources in an on-line environment," *INTED2009 Proceedings*, pp. 4312–4321, 2009.
- [10] M. Maćkowski, P. Brzoza, M. Żabka, and D. Spinczyk, "Multimedia platform for mathematics' interactive learning accessible to blind people," *Multimedia Tools and Applications*, pp. 1–18, 2017. doi: 10.1007/s11042-017-4526-z
- [11] I. Kohanova, "The ways of teaching mathematics to visually impaired students," in *International Congress on Mathematical Education (ICME)*, 2008.
- [12] S. Yang and Y. Ko, "Mathematical formula search using natural language queries," *Advances in Electrical and Computer Engineering*, vol. 14, no. 4, pp. 99–104, 2014. doi: 10.4316/AECE.2014.04015
- [13] M. Liška, P. Sojka, and M. Růžicka, "Similarity search for mathematics: Masaryk university team at the ntcir-10 math task," in *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies*. Citeseer, 2013, pp. 686–691.
- [14] M. Liška, P. Sojka, and M. Růžicka, "Math indexer and searcher web interface," in *International Conference on Intelligent Computer Mathematics*. Springer, 2014, pp. 444–448.
- [15] J. Mišutka and L. Galamboš, "Extending full text search engine for mathematical content," *Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008*, pp. 55–67, 2008.

TABLE III

SEARCH RESULTS FOR VARIOUS VISUAL QUERIES IN A GENERAL DATABASE WITH RESPECT TO FORMATS OF COMPARED DATA (INNER – LATEX, ENG – NATURAL ENGLISH, POL – NATURAL POLISH), AND THE MEASURE USED FOR STRING COMPARISON (L - LEVENSTEIN, COS - COSINE). LAST 3 COLUMNS PRESENT THE BEST 3 MATCHES.

Query	Data format	Measure	Match #1	Match #2	Match #3
$\int_1^2 \cos x dx$	inner	Lev	$\int_a^b f(x) dx$	$Y - y = \frac{dy}{dx}(X - x)$	$(a + b)^2$
	inner	cos	$\iiint_V \left( \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial R}{\partial z} \right)$	$\int_a^b f(x) dx$	$a_k = \frac{2}{T} \int_0^T f(x) \cos k\omega x dx$
	Eng	Lev	$\int_a^b f(x) dx$	$y' = P(x)y^2 + Q(x)y + R(x)$	$a_n = a_1 q^{n-1}$
	Eng	cos	$\int_a^b f(x) dx$	$a_k = \frac{2}{T} \int_0^T f(x) \cos k\omega x dx$	$\int_a^b y dx \approx h(y_0 + y_1 + \dots + y_{n-1})$
	Pol	Lev	$\int_a^b f(x) dx$	$(a + b)^2$	$y' = P(x)y^2 + Q(x)y + R(x)$
	Pol	cos	$\int_a^b f(x) dx$	$\int_a^b y dx \approx h(y_0 + y_1 + \dots + y_{n-1})$	$\int_a^b y dx \approx \frac{1}{2}h(y_0 + 2y_1 + \dots + 2y_{n-1} + y_n)$
$\sqrt[n]{a_n}$	inner	Lev	$(a + b)^2$	$Y - y = \frac{dy}{dx}(X - x)$	$\int_a^b f(x) dx$
	inner	cos	$\sqrt[n]{m\sqrt{x}} = m \cdot \sqrt[n]{x}$	$\frac{a_1+a_2+\dots+a_n}{n} \geq \sqrt[n]{a_1 a_2 \dots a_n}$	$\sqrt[n]{x_1 x_2 \dots x_n}$
	Eng	Lev	$\int_a^b f(x) dx$	$a_n = a_1 q^{n-1}$	$\sqrt[n]{m\sqrt{x}} = m \cdot \sqrt[n]{x}$
	Eng	cos	$\sqrt[n]{m\sqrt{x}} = m \cdot \sqrt[n]{x}$	$\sqrt[n]{\frac{a}{b}} = \frac{\sqrt[n]{a}}{\sqrt[n]{b}}$	$\sqrt[n]{x_1 x_2 \dots x_n}$
	Pol	Lev	$\sqrt[n]{m\sqrt{x}} = m \cdot \sqrt[n]{x}$	$\frac{a_1+a_2+\dots+a_n}{n} \geq \sqrt[n]{a_1 a_2 \dots a_n}$	$S_n = \frac{n(a_1+a_n)}{2}$
	Pol	cos	$\sqrt[n]{m\sqrt{x}} = m \cdot \sqrt[n]{x}$	$\sqrt[n]{x_1 x_2 \dots x_n}$	$\sqrt[n]{\frac{a}{b}} = \frac{\sqrt[n]{a}}{\sqrt[n]{b}}$
$\frac{x^2}{a^2}$	inner	Lev	$\sin z = \frac{e^{iz} - e^{-iz}}{2i}$	$\cos z = \frac{e^{iz} + e^{-iz}}{2}$	$\int_a^b f(x) dx$
	inner	cos	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = z$	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$	$(a - b)^2 = a^2 - 2ab + b^2$
	Eng	Lev	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = z$	$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$	$\sqrt[n]{\frac{a}{b}} = \frac{\sqrt[n]{a}}{\sqrt[n]{b}}$
	Eng	cos	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = z$	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$	$(a - b)^2 = a^2 - 2ab + b^2$
	Pol	Lev	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = z$	$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$	$\cos z = \frac{e^{iz} + e^{-iz}}{2}$
	Pol	cos	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = z$	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$	$(a - b)^2 = a^2 - 2ab + b^2$

[16] P. Sojka and M. Lřřka, "The art of mathematics retrieval," in *Proceedings of the 11th ACM symposium on Document engineering*. ACM, 2011, pp. 57–60.

[17] M. Adeel, M. Sher, and M. S. H. Khiyal, "Efficient cluster-based information retrieval from mathematical markup documents," *World Applied Sciences Journal*, vol. 17, no. 5, pp. 611–616, 2012.

[18] M. N. Quoc, K. Yokoi, Y. Matsubayashi, and A. Aizawa, "Mining coreference relations between formulas and text using wikipedia," in *23rd International Conference on Computational Linguistics*, 2010, p. 69.

[19] D. Formánek, M. Lřřka, M. Růřička, and P. Sojka, "Normalization of digital mathematics library content," in *Joint Proceedings of the 24th Workshop on OpenMath and the 7th Workshop on Mathematical User Interfaces (MathUI)*, 2012, p. 91.

[20] P.-Y. Chien and P.-J. Cheng, "Semantic tagging of mathematical expressions," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 195–204.

[21] M. E. Altamimi and A. S. Youssef, "Wildcards in math search, implementation issues," in *CAINE*. Citeseer, 2007, pp. 90–96.

[22] A. Niewiarowski and M. Stanuszek, "The mechanism of identification and classification of content (in Polish)," *Studia Informatica*, vol. 34, no. 2B, pp. 205–222, 2013.

[23] D. Połap, "Neural validation of grammatical correctness of sentences," *Ceur-ws*, 2016.

[24] S. Kozielski, M. řwiderski, and M. Bach, "The use of natural language as an intuitive semantic integration system interface," in *Internet-Technical Development and Applications*. Springer, 2009, pp. 51–58.

[25] J. Jagielski and P. Wnęk, "Natural language in databases systems," *Studia Informatica*, vol. 31, no. 2B, pp. 281–290, 2010.

[26] F. Li and H. Jagadish, "Constructing an interactive natural language interface for relational databases," *Proceedings of the VLDB Endowment*, vol. 8, no. 1, pp. 73–84, 2014.

[27] R. Alexander, P. Rukshan, and S. Mahesan, "Natural language web interface for database (nlwidb)," *arXiv preprint arXiv:1308.3830*, 2013.

[28] G. Navarro, "A guided tour to approximate string matching," *ACM Comput. Surv.*, vol. 33, no. 1, pp. 31–88, Mar. 2001. doi: 10.1145/375360.375365. [Online]. Available: <http://doi.acm.org/10.1145/375360.375365>