UNIVERSITÀ DEGLI STUDI DI TORINO

DIALOGARE CON LA MATEMATICA: VERSO UN'INTERAZIONE DIALOGICA VOCALE AUTOMATICA PER LA NAVIGAZIONE DI ESPRESSIONI MATEMATICHE

Presentazione a cura di:
Pier Felice Balestrucci

INDEX

CONTEXT

- Assistive technologies are those technologies that make IT products accessible and usable for people with disabilities
- The aim of this work is to make easier and more understandable to listen to mathematical formulas for visually impaired and blind people
- Mathematical formulas are rich in symbols that are difficult to be read by screen readers

OBJECTIVES

1. Starting from mathematical formulas generating mathematical sentences for the English

2. Combining the generation system with a dialogue system

3. Introducing the property of "navigability" into the system: the user can ask the system to repeat parts of the mathematical phrase

INDEX

HOW TO REPRESENT MATHEMATICS?

- Mathematics is less intuitive to represent than a simple text!

- The page space is exploited continuously with symbols arranged on multiple levels

$$\sqrt{x} - \frac{x+1}{x} = 0$$

LATEX

- Expression of the formula in latex:

$$\sqrt{x} - \frac{x+1}{x}=0$$

- Its representation:

$$\sqrt{x} - \frac{x+1}{x} = 0$$

CONTENT MATH MARKUP LANGUAGE

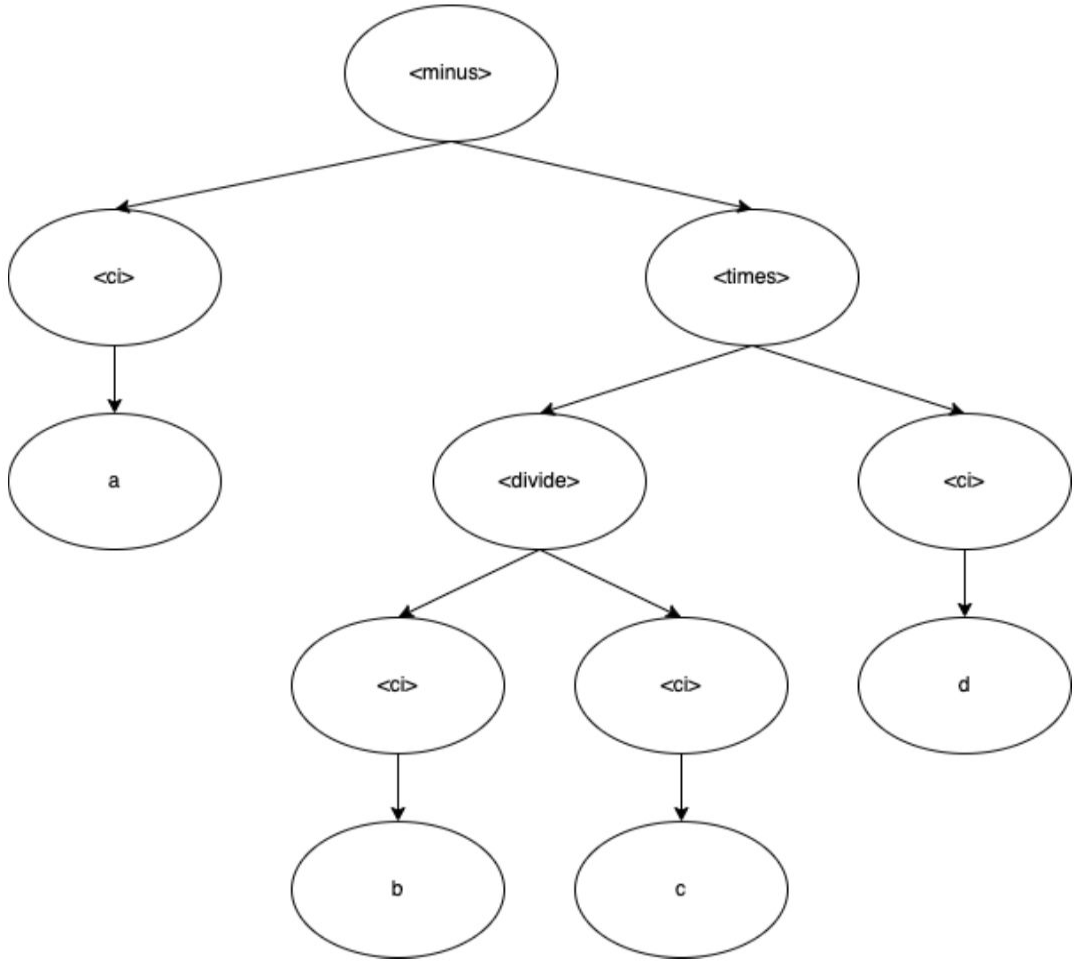
- CMML is a markup language useful for representing the semantics of a mathematical expression

- Look at the following formula $a - \frac{c}{b}d$:

```
<apply>
    <minus/>
        <ci>a</ci>
        <apply>
            <times/>
                <apply>
                    <divide/>
                        <ci>b</ci>
                        <ci>c</ci>
                </apply>
                <ci>d</ci>
        </apply>
</apply>
```

CONTENT MATH MARKUP LANGUAGE

- CMML is a markup language useful for representing the semantics of a mathematical expression
- Look at the following formula $a - \frac{c}{b}d$:



THE VARIOUS TYPES OF OPERATORS

Similar operators have similar linguistic structure and therefore we can divide them into various categories:

1. Relational operators
2. Algebraic, arithmetic and set operators
3. Logical Connectives
4. Conditional set
5. Pair
6. Auxiliary
7. Sequences
8. Elementary functions
9. Calculation

EXAMPLE: ALGEBRAIC, ARITHMETIC AND SET OPERATORS

Simbolo	Operatore	Forma Inglese
$+$	plus	plus
$-$	minus	minus
$/$	divide	over
$*$	times	times
$[\ldots]^{[..]}$	power	to
$\setminus$	settdiff	minus
$\times$	cartesianproduct	cross
$\cap$	intersect	the intersection of
$\cup$	union	the union of ... and ...

EXAMPLE: ALGEBRAIC, ARITHMETIC AND SET OPERATORS

- It is possible to determine two types of forms in this case: unary and binary:
  - For example, the "minus" operator can define an expression of the type a −b or act as an adjective by defining a negative value −5

For this reason, depending on the nodes that make up the operator, the realization has been made different

A GRAMMAR THAT GENERATES THIS TYPE OF OPERATORS

The first step is to determine a grammar, Context-Free, that can generate sentences for this type of operators:

$$
\begin{aligned}
&S \rightarrow NP\ VP \mid NP \\
&VP \rightarrow V\ NP \\
&NP \rightarrow N \mid Adj\ N \\
&PP \rightarrow P\ NP
\end{aligned}
$$

Example:

a) (minus, Adj) (5, N)

b) (a, N) (times, V) (b, N)

A GRAMMAR THAT GENERATES THIS TYPE OF OPERATORS

- Having determined the grammar, we compose the sentence:

a) **Unary form**: for this case a so-called "adjective phrase" is created and the operator plays the role of adjective

b) **Binary form**: constructs a sentence that has the operator (times) as its parent node and the two elements of the formula (a and b) as children. There are cases in which prepositions also appear: they are aggregated by placing them after the operator

SIMPLE NLG

- SimpleNLG is a Java library that gives you direct control over the process by which sentences are constructed, morphological operations, and linearization. The steps are:
  1. Sorting the various phrases in the tree according to their dependency bond
  2. Conjugating verbs at the specified tense
  3. Inflecting adjectives in general and number in accordance with that of the noun to which they are associated
  4. Inflecting names in general and number if explicitly requested to do so
  5. Choosing the correct version of the articles based on the name they are associated with
  6. Denying the sentence if explicitly requested to do so

INDEX

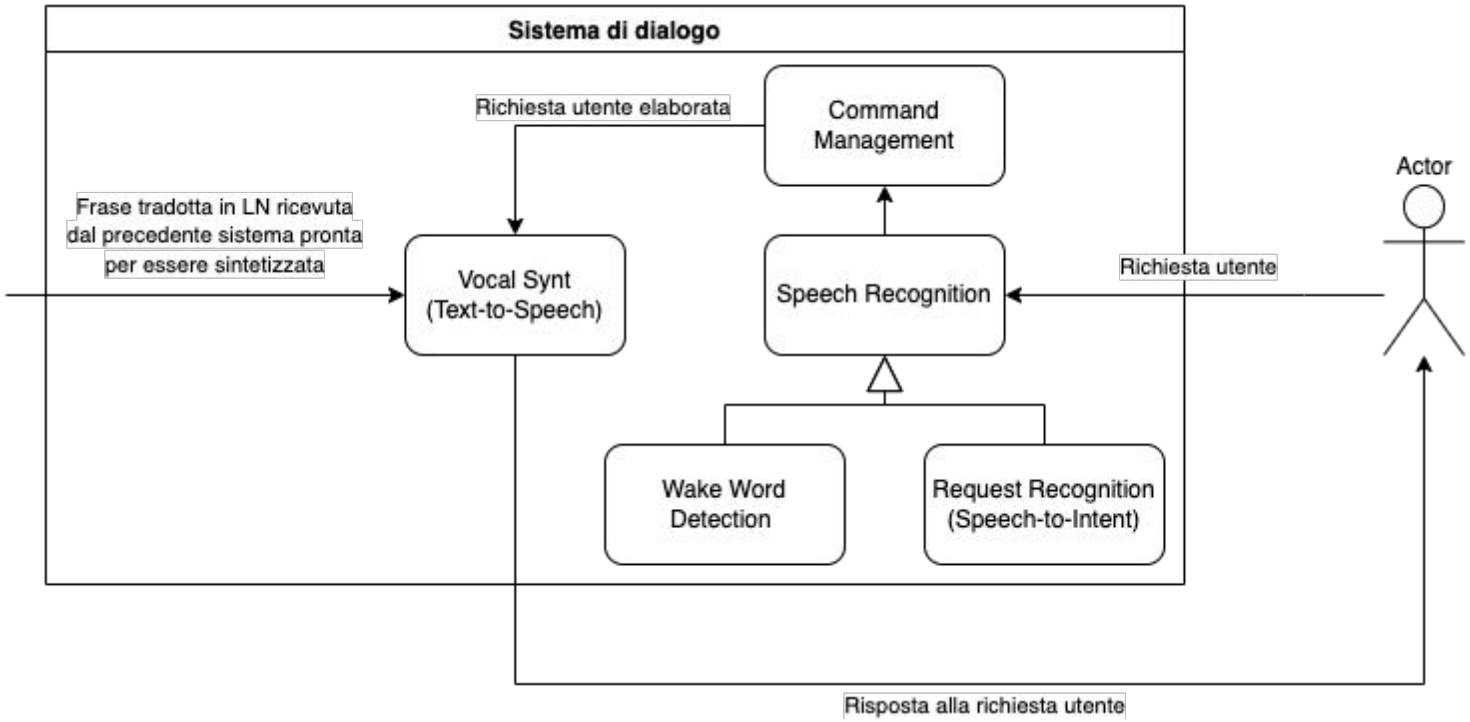
DIALOGUE SYSTEM

- Why a dialogue system?
  - To improve the understanding of mathematical phrases
- Its design had to take into account both the project objectives and the need to have sufficiently good standards of human-computer interaction

FEATURES OF THE DIALOGUE SYSTEM

1. It can be interrupted at any time via voice input
2. Understand and analyze user requests
3. Answer certain user questions

DIALOGUE SYSTEM



TEXT-TO-SPEECH

- Text-to-Speech (TTS) is a technology that converts text to speech output

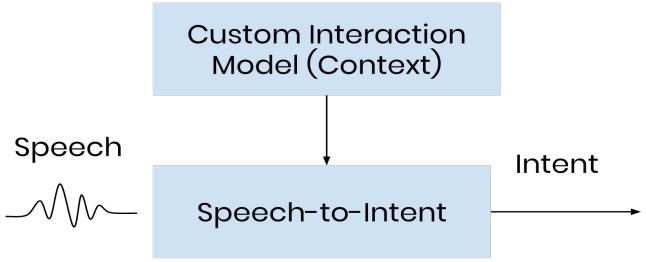
- The synthesizers used to voice the system are AWS Polly and eSpeak
  - **AWS Polly**: uses advanced deep learning technologies to synthesize human-like natural speech
  - **eSpeak**: is an open source software speech synthesizer based on the method of synthesis of formants

WAKE-UP WORD DETECTION

- When using modern dialogs such as Alexa, Siri or Google Assistant, users first interact with certain wake words
- They are binary classification systems that allow you to understand whether or not the user wants to interact with the voice assistant at that moment
- The wake up word is a default word that is fixed during training and detection

SPEECH-TO-INTENT

- Such systems work on a narrow vocabulary of terms and match the user request to an intention. This is called **intent classification**
- We categorize a user's intent, where a combination of intents allows a conversation to be defined
- Rhino is a Speech-to-Intent engine, which allows you to directly deduce intent from voice commands within a certain context in real time



COMMAND MANAGEMENT

- The last piece that completes the picture of the project concerns the management of the commands that the user can use

- From the moment the system listens to user requests, the user can ask two types of questions:
  1. Repeat from ...
  2. What is ...

ALGORITMO PRINCIPALE DEL SISTEMA DI DIALOGO

```
dialogue():
1.  Say "I’m starting to say the sentence"
2.  Activate the Wake Word Detection system
3.  For each word within the sentence:
4.      Say the word
5.      If the Wake-up Wor Detection system has detected "Ehy stop":
6.          Say "Ok, I’m listening to you"
7.          Activate the Request Recognition system
8.          Fulfill the request
9.  Say "I’ve finished reading this sentence, but I’m still here for you"
10. Activate the Request Recognition system
```

ALGORITMO DI GESTIONE DELLE RICHIESTE

```
fulfill_request(intent, request):
1.  Parsify the request
2.  If the intent is "Questioning":
3.      Search for the operator in the request
4.      Match the request with the operator arguments
5.      If the match is successful:
6.          Answer the question
7.          Otherwise, utter an apology message
8.  If the intent is "RepetitionFrom":
9.      Search within the sentence uttered what the user
                                        wants to hear repeated
10.     If the search is successful:
11.         Repeat from that point
12.         Otherwise, utter an apology message
13. If the intent is "Resume":
14.     Resume the synthesis of the mathematical expression
```

INDEX

EXPERIMENTATION

In order to test the entire system, it was decided to divide the experimentation into two phases:

1. The first is designed to test the effectiveness of the mathematical sentence generation system:
   - There were two blind users who took the test: both native Italian speakers with excellent knowledge of mathematics
2. The second to test both the effectiveness of the dialogue system and its usability:
   - There were five testers: all visually impaired as native Italian speakers with generally sufficient mathematical knowledge

FIRST EXPERIMENTATION PHASE

For this phase, a questionnaire divided into two parts was administered:

- In the first, profiling questions are asked
- In the second, instead, you are asked to listen to 10 formulas, through videos uploaded to Youtube, summarized according to different parenthesization strategies and for each one you are asked to:
  - Rewrite the formula heard using an unambiguous notation
  - Answer a multiple-choice question formula on a seven-value Likert scale in order to evaluate the understanding of the formula

EASY FORMULAS

Formula LaTeX	Nodi
$A \times B = \{(x, y) \mid x \in A, y \in B\}$	15
$g^{-1}(y) = f^{-1}\left((y-b)/a\right)$	13
$\int_b^c a \, \mathrm{d}x = a(c-b)$	14
$x > b \Longrightarrow \lvert f(x)\rvert < M$	10
$\sqrt[n]{x} = x^{1/n}$	10

DIFFICULT FORMULAS

Formula LaTeX	Nodi
$\lim_{x \to x_0} \left\{ \frac{f(x) - f(x_0)}{x - x_0} - f'(x_0) \right\} = 0$	31
$y = f(a) + \frac{f(b) - f(a)}{b - a}(x - a)$	21
$\int \frac{1}{\sqrt{m^2 - x^2}} \mathrm{d}x = \arcsin \frac{x}{m} + c$	20
$\sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$	28
$\lim \left(1 + \frac{1}{n}\right)^n = e$	10

EXACT MATCH RESULTS

Utente	Tot. formule (25)	F. facili (10)	F. difficili (15)
1	23	10	13
2	25	10	15
Tot:	48(96%)	20(100%)	28(93%)

Exact Match is a metric that returns a value of 1 (or 0) if the initial and perceived CMML trees are equal (or not)

SPICE RESULTS

Utente	Tot. formule (25)	F. facili (10)	F. difficili (15)
1	0.98	1.0	0.97
2	1.0	1.0	1.0
AVG:	0.99	1.0	0.985

The SPICE score is a measure of similarity that is obtained by calculating the F-score of the overlap between the two CMML trees. The overlap is measured by decomposing the trees into typed elementary substructures (operands, operators and their relationships). For example, the expression $x - 1$ is decomposed as 1, x, minus, (op : minus, prime : x), (op : minus, second : 1)

SECOND EXPERIMENTATION PHASE

For the second phase, test users had to respond to both requests to verify the usefulness of the tool and questions to evaluate the User Experience

- Users were asked both profiling questions and to compile the User Exceperience Questionnaire

EASY FORMULAS

Formula LaTeX	Nodi
$g^{-1}(y) = f^{-1}\left((y-b)/a\right)$	13
$\int_b^c a\,\mathrm{d}x = a(c-b)$	14
$\sqrt[n]{x} = x^{1/n}$	10

DIFFICULT FORMULAS

Formula LaTeX	Nodi
$\lim_{x \to x_0} \left\{ \frac{f(x) - f(x_0)}{x - x_0} - f'(x_0) \right\} = 0$	31
$y = f(a) + \frac{f(b) - f(a)}{b - a}(x - a)$	21
$\int \frac{1}{\sqrt{m^2 - x^2}}\mathrm{d}x = \arcsin\frac{x}{m} + c$	20

EXACT MATCH RESULTS

Utente	Tot. formule (6)	F. facili (3)	F. difficili (3)
1	3	2	1
2	6	3	3
3	2	1	2
4	2	0	2
Tot:	13(54.2%)	6(50%)	8(66.7%)

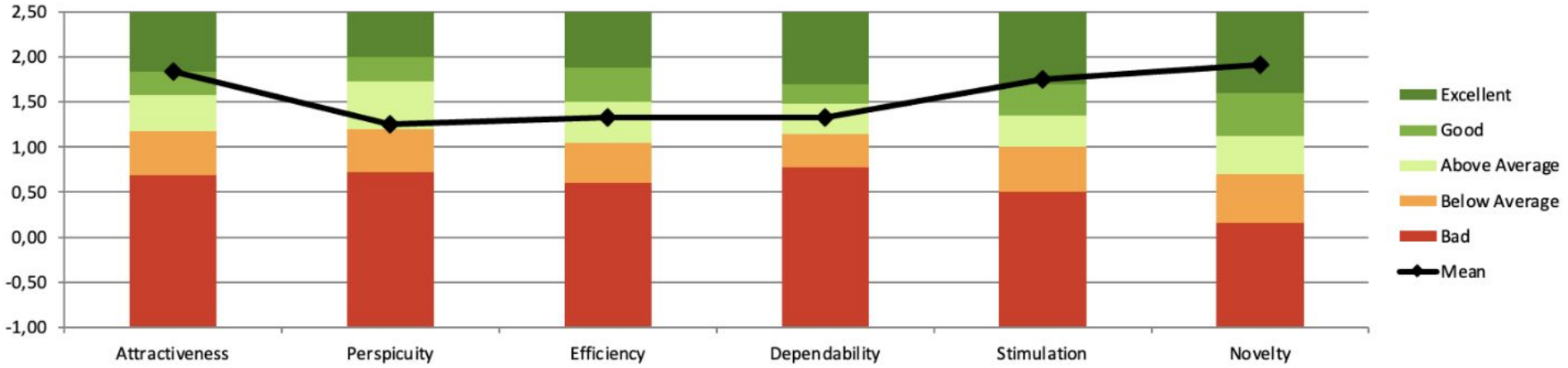
Exact Match is a metric that returns a value of 1 (or 0) if the initial and perceived CMML trees are equal (or not)

SPICE RESULTS

Utente	Tot. formule (6)	F. facili (3)	F. difficili (3)
1	0.86	0.89	0.87
2	1.0	1.0	1.0
3	0.66	0.98	0.82
4	0.80	0.95	0.89
AVG:	0.83	0.95	0.89

The SPICE score is a measure of similarity that is obtained by calculating the F-score of the overlap between the two CMML trees.
The overlap is measured by decomposing the trees into typed elementary substructures (operands, operators and their relationships).
For example, the expression $x - 1$ is decomposed as 1, x, minus, (op : minus, prime : x), (op : minus, second : 1)

UEQ RESULTS



Benchmark summarizing the behavior of the system against six different scales of UEQ:
attraction, perspicuity, efficiency, reliability, stimulation and novelty.
These scales measure both the aspects of classic usability (efficiency, perspicuity, reliability) and those of user experience (originality, stimulation). The attraction, on the other hand, is an indication of the overall impression of the product.
Shows the average values obtained in relation to a pre-existing reference dataset

INDEX

CONCLUSIONS

- The results produced show how the application successfully transforms mathematical formulas into natural language sentences in English

- Their high comprehension is an indication of both the ability of the system to synthesize mathematical sentences in English and the effectiveness of the synthesizers used.

CONCLUSIONS

- The results obtained from the second phase of the experimentation show that the dialogue system is very good, but it can be significantly improved
- The feedback, however, left by users was very positive and a sign of an enthusiasm dictated by the news and possibilities that such a system can give them

POSSIBLE FUTURE DEVELOPMENTS

Extending the set of operators managed by the language system and generation of mathematical phrases in English

Expanding the system so that it is able to translate also mathematical graphs and chemical formulas

Improving the dialogue system by introducing new features such as the ability to make users set the speed and voice of the speaker

Expanding the voice commands of the dialogue system and its understanding to users' questions

Reshaping the dialogue system by formalizing interaction on speech acts

NOW IN DEVELOPING…

Expanding the system so that it is able to describe and navigate graphs such as DFA

THANKS FOR YOUR ATTENTION!