



7th International conference on Intelligent Human Computer Interaction, IHCI 2015

Initial implementation of natural language turn-based dialog system

Alexander Wachtel*, Sebastian Weigelt, Walter F. Tichy

Karlsruhe Institute of Technology, Chair for Programming Systems Prof. Walter F. Tichy, Am Fasanengarten 5, 76131 Karlsruhe, Germany

Abstract

Our prototype implements a natural language dialog system for Excel spreadsheets. The work is motivated by a pilot study which shows that novice users have difficulties with the formula language of Excel and need interactive assistance. JustLingo¹ allows spreadsheet calculations in ordinary, written English. Furthermore, users can extend the system functionality by explaining of the calculations to the prototype. The keyword-based system analyzes the input using natural language processing techniques and translates it into Excel formulas. It asks for missing information such as operands and target cells and provides alternatives if there are ambiguities. It also handles references to previous inputs, allowing step-by-step construction of calculations. An evaluation shows that it properly resolves 82 % of references and correctly interprets 79,5 % of overall user input. Although far from perfect, the prototype demonstrates that computers may soon begin to take on simple, limited-domain programming tasks based on natural-language input.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Organizing Committee of IHCI 2015

Keywords: Natural Language Processing; Natural Language Interfaces; Dialog Systems; End-User Programming; Spreadsheet

1. Introduction

Since their invention, digital computers have been programmed using specialized, artificial notations, called programming languages. However, only a tiny fraction of human computer users can actually work with those notations. An alternative would be ordinary, natural language. Ordinary language would enable almost anyone to program and would thus cause a fundamental shift in the way computers are used. Rather than being a mere consumer of programs written by others, each user could write his or her own programs². The idea of programming in natural language was first proposed by Sammet in 1966¹, but enormous difficulties in this area have resulted in progress being disappointingly slow. In recent years, significant advances in natural language techniques have been made, leading, for instance, to IBM's Watson³ computer winning against the two Jeopardy! World champions, Apple's Siri routinely answering wide-ranging, spoken queries, and automated translation services such as Google's becoming usable. However, programming in natural language remains an open challenge^{4,5}.

¹ <http://ps.ipd.kit.edu/english/JustLingo.php>

* Corresponding author. Tel.: +49-721-608-46320 ; fax: +49 721-608-42817.

E-mail address: Alexander.Wachtel@kit.edu

The first prototype allows users to express spreadsheet calculations in ordinary typed English and converts the natural language input into arithmetic spreadsheet formulas. In general, spreadsheets have been used at least 7000 years for information representation and correlations within the data⁶. Meanwhile, spreadsheet programs such as Excel Microsoft become ubiquitous. It is estimated that each year hundreds of millions of spreadsheets are created⁷. A small knowledge base and an interactive dialog system assist in the accomplishment of computational tasks. The approach uses a limited vocabulary and a small set of syntactic patterns organized as a grammar. Relying on 92 patterns covering all basic arithmetic operations, both syntactic and semantic analysis can be performed simultaneously. This limited range of possibilities enables us to derive most of the semantics from the user query precisely. By recognizing references to previous inputs the system supports step-by-step construction of complex expressions. It also requests missing information and asks the user for clarifications in case of ambiguities. Rather than relying on the completeness of particular inputs, the system establishes a context which makes it usable for inexperienced users.

Perhaps the latter insight is the most significant: In the past, computers were expected to follow instructions blindly, without a notion of right or wrong or what users expected. Natural language programming, on the other hand, requires a domain-aware counterpart that can ask for clarification, thereby overcoming the chief disadvantages of natural language, namely ambiguity and imprecision.

2. Pilot study

Initially, we needed to find out how real users would interact with spreadsheets in natural language. For this purpose, we recruited subjects and asked them to describe how to solve problems with the aid of a spreadsheet. Spreadsheets with data were provided together with the tasks. The subjects were asked to imagine explaining to a human partner how to solve a given task and to write down what they would say. Even though there were no suggestions to ask the imaginary partner for help, many participants did just that. The problems to be solved were selected from the first four chapters of the textbook Excel 2013 Step by Step⁸. The study consists of 35 questions, six of which are mathematical problems (i.e. calculation of sum and average, rounding decimal values, copying an existing formula to other cells, conditional functions) and another six are data and document manipulation tasks (i.e. inserting and sorting of columns, creating tables and diagrams, labeling cells).

2.1. Observations

Overall, there were 57 participants. In a self-assessment, 12 % considered themselves as experts, 72 % advanced users, and 16 % beginners with regard to spreadsheets. The participants' experience on working with Excel was on average 9.5 years, with a maximum of 25 years and a minimum of 1 year. The following discusses a sample scenario from the empirical study that involves arithmetic functions within an Excel worksheet (see Figure 1).

	A	B	C	D	E	F
1	Grades overview					
2						
3	Name	Daily grades 1	Daily grades 2	Daily grades 3	Final exam	Average
4	Marcel	35,7	35,3	20,0	40,0	
5	Andrea	75,0	95,0	65,0	95,0	
6	Simon	100,0	95,0	75,0	85,0	
7	Andrew	90,0	95,6	60,0	80,8	
8	Mike	100,0	35,0	70,0	45,0	
9	Tyler	95,5	100,0	85,1	90,0	
10	Anna	60,0	40,0	35,0	35,0	
11	Mary	80,0	85,0	80,0	90,4	
12	Flinn	65,0	20,0	70,9	40,6	
13	Rick	30,3	20,7	40,0	20,0	
14	Alex	80,0	60,0	85,0	20,0	
15	Average	73,77	61,96	62,36	58,35	

Fig. 1: Example scenario in spreadsheet

The problem to be solved was stated as follows: In the column "Average" the mean of all 4 grades should be calculated for each student. Only the average value regardless of the weighting mentioned before should be calculated. Your task is to perform that calculation specifically for the student Marcel and to write it to F4.

Download English Version:

<https://daneshyari.com/en/article/488543>

Download Persian Version:

<https://daneshyari.com/article/488543>

[Daneshyari.com](https://daneshyari.com)