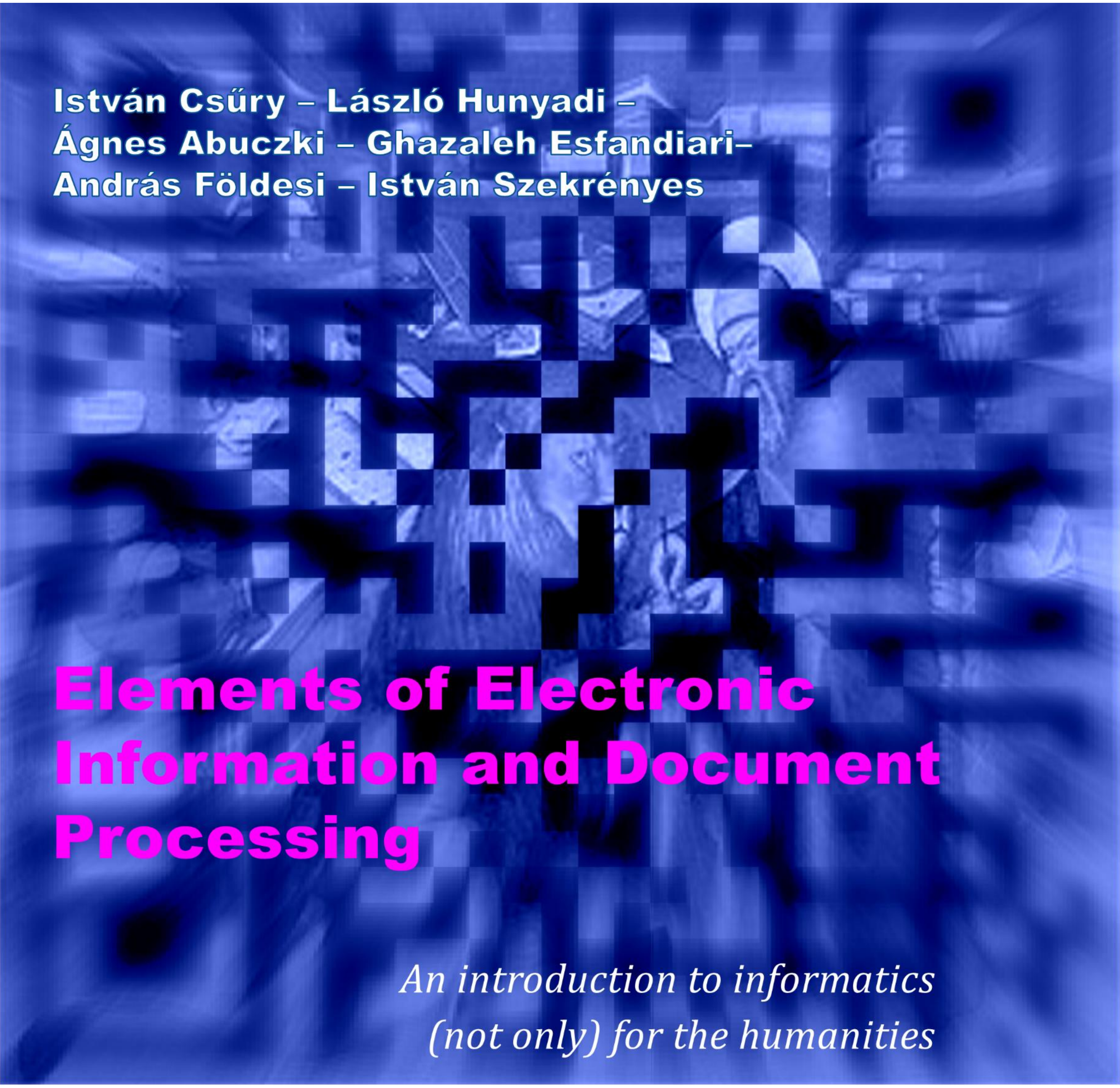


István Csűry – László Hunyadi –
Ágnes Abuczki – Ghazaleh Esfandiari–
András Földesi – István Szekrényes

Elements of Electronic Information and Document Processing

*An introduction to informatics
(not only) for the humanities*

edited by István Csűry

TÁMOP-4.1.2.D-12/1/KONV-2012-0008

Debrecen University Press
2016

SZÉCHENYI 2020

Európai Unió
Európai Szociális
Alap

MAGYARORSZÁG
KORMÁNYA

BEFEKTETÉS A JÖVŐBE

István Csűry – László Hunyadi –
Ágnes Abuczki – Ghazaleh Esfandiari– András Földesi – István Szekrényes

Elements of Electronic Information and Document Processing

An introduction to informatics
(not only) for the humanities

edited by István Csűry

Debrecen University Press

2016

Edited by István Csűry
Reviewed by István Károly Boda

This publication was supported by the TÁMOP-4.1.2.D-12/1/KONV-2012-0008 project („Szak-nyelv-tudás” – Az idegen nyelvi képzési rendszer fejlesztése a Debreceni Egyetemen)

Textbook available in electronic format only

ISBN 978-963-318-564-3

DU press
Kiadta a Debreceni Egyetemi Kiadó 2016-ban – Published by Debrecen University Press, 2016
www.dupress.hu
Felelős kiadó: Karácsony Gyöngyi

Table of Contents

Ingredients for efficiency

(INSTEAD OF AN INTRODUCTION[i])

István Csűry

Are you just wondering how to tackle a fastidious task you are assigned to or how to realize some absolutely original idea that came to your mind? Perhaps there is already a special computer program out there responding exactly to your needs, or at least capable of making your job easier (to say nothing of tasks that are simply unrealizable manually). There are software applications for (almost) everything; you only have to look for it over the Internet. Maybe it takes some time to find the right tool and to become familiar enough with it, but it is worth the effort, unless you prefer to spend all your time on working (at a low efficiency rate, and without being sure of a consistent and faultless result). It may seem obvious, but people reluctant to learn, for instance, the use of formulas and functions in Excel, even should it cost them to work after hours regularly, are far from being an exception. Discovering new tools seems then an even bigger challenge. But let us give you an encouragement. *Content management system* is not much of a pleasant-sounding expression, is it? However, could anyone pretend that s/he has never used some *content management system* (in a broader sense of the term)? Not plausibly if s/he has ever posted a single comment on a web site, used Facebook or kept a blog. Of course, there are more demanding tasks in handling such a system, but one of its main purposes is to allow users with no knowledge about HTML, computer networks and programming to contribute to the creation of web sites and to publish contents there.[ii] Generally speaking, most of computer programs are not for informaticians, programmers and computer geeks but for simple users in need of tools for their everyday work.

As a translator, a teacher or a researcher, you will want to carry out various operations on any kind of media. Transform a photographed document into an editable text[iii] or add neatly presented syntactic trees or other graphs to your papers[iv]. Manage bi- or multilingual lists of technical terms or edit some other vocabulary or dictionary.[v] Search for examples of how a word or a structure is employed in actual language use[vi] or compile and annotate a corpus[vii] allowing you a comprehensive analysis of (linguistic, literary, rhetorical, psycho-social, etc.) phenomena you are interested in. Realize statistical analyses of your data.[viii] Edit sound[ix], image[x] or video[xi], add or edit subtitles[xii], or convert a file from one format into another[xiii]. Or anything else: it is impossible to enumerate all possible tasks that can be facilitated by some software application. The following chapters illustrate some fields of activities in which computing is a key factor. Hopefully, they will also give readers a liking for searching efficient tools for whatever kind of work they have to do.

The present volume is addressed principally to students in humanities. It was inspired by computing-related courses the authors taught in various programs at the Faculty of Arts and Humanities of University of Debrecen as well as by their intention of giving undergraduate students some insight into natural language processing research activities[xiv].

As a common observation, we found that more effort should be spent on increasing computer literacy, especially in order to enable students to use computer skills for professional purposes. Obviously, there would be too many applications to be presented for a book intended to be concise. We often have to choose among a number of similar programs. Also, technology evolves at such a pace that books like this could not keep up if they would give concrete instructions on how to carry out concrete operations with concrete software. Therefore, we preferred to lay out an overview of some fields of computer applications that may be of a common interest for students in humanities.

Our book provides orientation, and aims to develop a conscious and creative attitude to use computer tools for any professional purpose, which is a basic element of this universe where self-teaching[xv] has an essential role. It is written in English, this lingua franca indispensable for understanding and utilizing the larger part of somewhat more specialized computer programs. Even though software localization has become common practice, special tools for special purposes are mostly available only in English. “Better get used to it” – or take the initiative to translate them yourself, as it is done in the case of community-developed software.

The book is composed of three chapters, according to three main activities studied in the framework of our training programs: research, teaching, and translation. The first (and longest) one encompasses all the stages of research process, from planning through organizing and data collection to analysis and preparing publications. It presents not only suitable tool types but also illustration with examples of corpus-based linguistic/pragmatic research. The second chapter deals with software-aided course authoring and exercise/test development, and resumes how, from a technical point of view, a virtual learning environment can be created and managed. The last chapter explains distinctions and convergences between computer-assisted and automatic solutions for translation, and summarizes essential facts about the latter. Definitions of basic notions as well as explanations on the computerized workflow of “industrial” translation enable the reader to understand phenomena about translation that we encounter even in everyday life.

One might ask why publication itself is not entirely covered, given the lack of a chapter on document editing and publishing tools, like word processors, the most banal instruments of each evoked activity. Text editors (just like spreadsheet or presentation editing software) are considered belonging to elementary computer literacy so that it seemed possible to us to cut corners at this point. Maybe many users are unfamiliar with some more advanced functionalities of these tools, like styles, templates, cross-reference, tables of contents, mail merge, review, track changes or comments. Might you be one of them? Well, it is time then to initiate that self-teaching procedure we have just mentioned. Otherwise, you are likely to waste endlessly your precious time on boring mechanical operations – with poor results.

[i] Do not worry about notes. You can read them all after the main text.

[ii] "A web content management system (WCMS) is a software system that provides website authoring, collaboration, and administration tools designed to allow users with little knowledge of web programming languages or markup languages to create and manage website content with relative ease." *(Wikipedia)* Examples of CMSs: Drupal (www.drupal.org), Joomla! (www.joomla.org), WordPress (www.wordpress.org) or MediaWiki (www.mediawiki.org).

[iii] Optical character recognition (OCR) is the automatic conversion of images of (usually typed or printed) text into computer-editable text. Examples of OCR systems: built-in OCR function in MS OneNote, MS Document Imaging, FreeOCR (www.freeocr.net) or Abbyy (www.abbyy.com).

[iv] Examples of such graphical editors: Dia Diagram Editor (http://dia-installer.de) or TreeForm (http://sourceforge.net/projects/treeform).

[v] Examples of dictionary-making tools: Toolbox Dictionary Factory (http://www-01.sil.org/computing/toolbox/techniques.htm) or WeSay (http://wesay.palaso.org/).

[vi] "A concordancer is a computer program that automatically constructs a concordance" *(Wikipedia)*, i.e. a list of words or utterances of a given word in a corpus of texts according to search criteria determined by the user. Examples of concordancers: AntConc (www.laurenceanthony.net/software/antconc) or Tom Cobb's online concordancers on the Compleat Lexical Tutor web site (www.lextutor.ca/conc).

[vii] Example of such a tool: UAM CorpusTool (www.wagsoft.com/CorpusTool).

[viii] "Statistical software are specialized computer programs for statistical analysis and econometric analysis." *(Wikipedia)*. Examples of statistical software: SPSS (http://www-01.ibm.com/software/analytics/spss) or GNU PSPP (www.gnu.org/software/pspp).

[ix] An example of a sound recorder and editor: Audacity (http://audacityteam.org)

[x] Examples of image editors: MS Office Picture Manager, Adobe Lightroom (https://lightroom.adobe.com) or Picasa (https://picasa.google.com)

[xi] Examples of video editors: MS Windows Movie Maker or Filmora video editor (http://filmora.wondershare.com/video-editor)

[xii] An example of a video subtitle editor: Subtitle Workshop (http://subworkshop.sourceforge.net)

[xiii] An example of an audio file converter: fre:ac (www.freac.org). An example of a video file converter: Any Video Converter (www.any-video-converter.com/products/for_video_free).

[xiv] Especially in the HuComTech project (http://metashare.nytud.hu/repository/browse/hucomtech-multimodal-corpus-and-database/80230f6e6ba811e2aa7c68b599c26a066e7e04f01c6043b485f6bf2f65945880 ; http://lingua.arts.unideb.hu/hucomtech-database/ ; see also: Hunyadi, L. □ Bertók, K. □ Németh T., E. □ Szekrényes, I. □ Abuczki, Á. □ Nagy, G. □ Nagy, N. □ Németi, P. □ Bódog, A. 2011. The outlines of a theory and technology of human–computer interaction as represented in the model of the HuComTech project. In: CogInfoCom Conference Proceedings, Budapest. 1-5.)

[xv] The main sources of knowledge for developing skills with software applications are "Help" menus, tutorials and other support found on the Internet, online community interactions and users' forums as well as personal experiments. In other words, an individual discovery procedure is always required. Fortunately, a simple search is enough in most cases to quickly bring us clear step-by-step instructions on particular operations. Too many users do not consider this kind of solution.

1

Documentation and Research

Mind Mapping, Project Management and Referencing

István Csűry

While the three types of tools mentioned in the title seem at first not to be intrinsically linked and conceived not essentially for specialists in humanities, they are all useful if not indispensable accessories for managing research and publication projects. These tools improve efficiency in processing specific data at critical points of workflow. Moreover, as we will see considering the examples, some software offer a combination of these basic tool types (for instance, MindManager is a mind mapping and a project management tool at the same time, and Docear combines mind mapping with referencing).

1) (Not only) mind mapping

i) What is mind mapping and what are related software good for?

A short overview

Text is linear, at least on the surface, but the world and thoughts it refers to are not. Not surprisingly, linearity determines general-purpose text editors and word processors. This may become a serious obstacle for *representing complex, multidimensional structures* and various kinds of *relations*, especially for visual-minded people. Pedagogical, psychological and even philosophical aspects of this issue are known long ago and have been widely discussed. What we are interested in is rather the technical aspect, the use of mind mapping software.

Graphically, a mind map is a kind of *spider diagram* that is made in several colours on a landscape oriented paper sheet and contains text labels, symbols and other graphical objects representing structured information. Its branches originate from a central notion, concept or category that they break down into components or aspects, represented by nodes. Nodes are organized in a strictly hierarchical structure; however, their groups as well as relations between distant nodes may be marked. Such a figure is suitable for obtaining a global yet analytic view of a given topic or problem. For example, when you are given an explanation during a lecture, or you are setting out the draft of a coursework, you may happen to note it spontaneously in the form of a mind map, even without knowing much about the methodology of mind mapping.

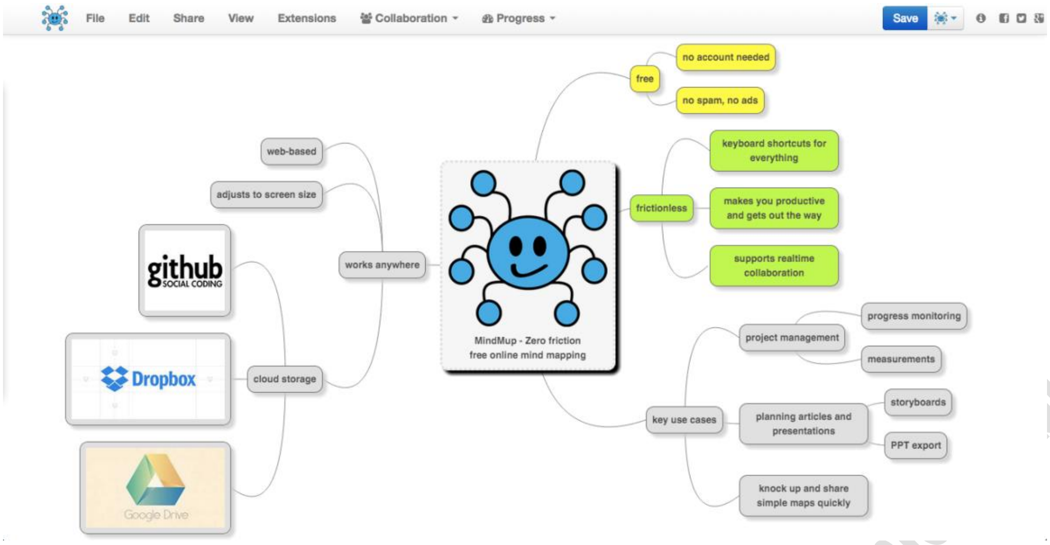


Figure 1 ***A simple mind map (by MindMup)***

Mind mapping software are not simply tools for editing such diagrams by computer instead of drawing them by hand. In addition to obvious advantages of software use on hand editing (editability, reusability, etc), they may allow *a better integration* of the work phase giving rise to a mind map *to the overall workflow*. Moreover, unlike other tools one can use for drawing diagrams, they are optimized for executing the specific tasks with the specific purposes described above.

Usual/Possible uses of mind mapping software

Although mind mapping software are not too difficult to learn, you would use it for *taking notes* (while attending lectures, meetings, etc) only when you got familiarized even with the hotkeys and shortcuts of the most frequent commands in order to be able to keep up with the speakers' speed. Then, you will appreciate its functionalities when feeding your notes into further working process.

Otherwise, a mind mapping software is *the tool for the initial stage of any kind of project*. It proves to be very helpful even for beginners in *brainstorming* a topic, *collecting and organizing information*, thoughts and ideas, *splitting them into components and subtasks* and, at the end, *lay out the plan* of the project, presentation, course, publication or whatever the nature and the aim of the work ahead of us might be.

In any case, the result (which is, informatically speaking, a file with a specific extension, based often on XML) can usually be *linked with other applications* (e. g. a calendar or a project management tool) and can be *exported* into various other formats allowing publication, sharing or further editing as a more conventional type of document.

ii) Main functionalities of mind mapping tools

The principal functionality of a mind mapping software is *diagram editing*: *creating branches (arcs) and nodes with text labels* (or bullets containing longer text) and modifying their structure as needed. Horizontal, i. e. hierarchical, and vertical *(re)ordering of nodes* is an easy task. For an easier work with large mind maps, branches can be *collapsed or expanded* at any time. *Formatting options and styles* at our disposal allow to visually emphasize structural relations and to weigh items according to their importance.

Depending on the software, some *project management utilities* may also be available. Solutions span from simple calendar options, that enable setting deadlines and reminders for tasks represented as nodes, to full integration of project management functionalities, i. e. collaboration solutions, assigning tasks to people in the project, etc. The mind mapping tool may then become a part of a software suite.

Most mind mapping tools feature the possibility of adding various kinds of *graphics* to our charts or *embedding multimedia* for quick reference. An option to *create hyperlinks* points toward the integration of mind maps with other software (e. g. web browser, file browser, office tools). *Exporting* and *publishing* mind maps can be seen as a minimal level of integration. In fact, native mind map files can usually be visualized only with the same software they were created with, which may be a hard limitation of usability. Therefore, even the simplest software of the category allow exporting mind maps in some common graphic file format, pdf often being an option as well. These formats still have limited functionality. More advanced solutions consist in enabling creation of web (html) pages, word processor/document editor files or even (animated, PowerPoint®- or Prezi®-like) presentations out of mind maps.

If we want to use mind maps as a part of teamwork, it is crucial to see whether and how we can share them. Web-based applications usually have got this feature while others offer integration with general-purpose web services and tools of cloud computing. Mind mapping software designed (also) for corporate use have built-in collaboration functionalities.

iii) Using the software

Everyone has to make his own choice among the many mind mapping software following his needs and preferences, probably after trying more than one; therefore, despite their similarities, we cannot offer a universal "user guide". Let us, however, provide the reader with some useful hints regarding certain functions and operations.

First of all, remember that, in the case of mind maps, *editing is not simply a word processing (or drawing) task*. Texts (as other data) belong to particular points of a hierarchical structure, so you have always to decide what you want to edit: the elements of the structure themselves or only their content. For instance, copy/cut/paste commands may apply to both, but with clearly different results. Similarly, you should be careful with entering text after a simple click on a node: your software may behave in this respect just like a spreadsheet editor and overwrite its content unless you have "opened" it for editing.

As for speeding up your editing of mind maps, the very first *shortcuts you should learn* are those that trigger the following actions:

- add a single node (below or above an existing one),
- add a new branch to a node, i. e. create child nodes,
- splitting and merging nodes,
- move nodes up or down with respect to their siblings,
- promote or demote nodes (often by cutting and pasting them), and
- edit node content.

Mind maps vary in dimension and structure from very tiny and simple to huge and intricate ones. Obviously, a more complex topic or a large project could not be represented in a simplistic way. However, too large, complicated/elaborated mind maps with too much detail are difficult to handle and, what is more, may fail the principal aim of mind mapping: making information more accessible. Thus, in such cases, consider *linking together simpler mind maps* by elaborating subcategories or subtasks of the principal one on further individual mind maps. Remember that your tool should allow you to embed a mind map in another as well as to export branches as new mind maps, or at least to hipertextually link several files.

Exporting is a critical task since mind mapping software are not as commonly used as web browsers, word processors or imaging applications. Mind maps often represent only the first step of a longer document editing process, too. Therefore, it is essential for us to be able to export our mind maps as image files, web pages, or text files that we can use as hierarchically structured drafts for further elaboration. Besides this criterion, integration with other tools (like an office software suite) as well as collaborating and sharing possibilities may also come into consideration when choosing a mind mapping program.

iv) Some examples of mind mapping software

name of software	author / publisher / company; website	main features
Freeplane	Dimitry Polivaev and others; http://freeplane.sourceforge.net/	• free, general purpose feature-rich mind mapping software • a fork of FreeMind • standalone application • no online editing or sharing in its default form • possibility of enhancements with add-ons (including collaborative work) • advanced functions like formulas and scripting
Docear	Information Science Group, University of Konstanz; http://www.docear.org/	• free and open source academic literature suite for information management and drafting academic writing • another fork of FreeMind • standalone application • mobile support and real-time collaboration under development • integrated reference, annotation and document management • includes a recommender system as well as a PDF metadata extraction and retrieval tool based on very large databases • powerful search & filter function • integration with MS-Word via add-on

name of software	author / publisher / company; website	main features
MindMup	Sauf Pompiers Ltd.; www.mindmup.com	• “zero-friction” free online mind mapping tool • sharing and online collaborative editing features • multiple versions: ○ online editing interface requiring no registration ○ Chrome extension (either online or offline) ○ “Gold” version (optional commercial service concerning storage and copyright) ○ Apple mobile version • compatibility with FreeMind (and, therefore, with Freeplane) • integration with public storage services (Google Drive, Dropbox…) • various importing and exporting possibilities • possibility of sharing maps optimized for posting to social networks and embedding in web sites through MindMup Atlas, a cloud mind map library • allows creating presentations from mind maps • measurements and project management features
MindManager	Mindjet LLC; http://www.mindjet.com/	• commercial software with free trial • mainly for business purposes • advanced project planning and task management features • budgeting and forecasting functions • allows creating presentations from mind maps • strongly integrates with office software • collaboration possibilities • adapted to mobile platforms

2) Projects and project management

i) What does *project* mean in software?

Project refers to some *more complex work involving multiple work phases, participants and/or objects* (e.g. files). It is a polysemic term. In computing, it appears frequently with a specific meaning in the menus of various kinds of software. For example, if you make a video with an appropriate software, you will want to use several footages, music, transition effects, text, etc., and all the files containing these data as well as the information about the way you want to put it all together will constitute a project. The program will thus make for you a project file describing all these, and, at the end, you will be able to create from your project a video file you can play on any device. Similarly, when we work in linguistics or discourse analysis with corpus annotation and analysis software, corpus files are not the only ones we have to handle: we usually have also annotation schemes, annotations and analyses, the ensemble of which being a project. Physically, it means that, for a given work, *several files (with one among them containing metadata about the whole) are created in a separate folder or structure of folders*.

In a more general sense, a project is "*an undertaking requiring concerted effort*" (The Free Dictionary by Farlex). Its work phases may also involve activities without any use of computer. The term is used in this sense as well in computing. Some programs contain functionalities for managing such types of projects, and there is a particular software type for *project management*.

In the following sections, we deal with project management in the latter, more common sense.

ii) What are project management software good for?

Project management software are powerful tools enabling users to face the difficulties even of the most complex project from the very beginning. Firstly, they are used in the initial stage for setting up the project plan in the detail. During the phase of execution, they are useful for monitoring and controlling the processes in such a way that adjustments could be made for the project being successful. Finally, the systematic use of a project management tool provides us with data necessary for evaluation purposes. At the same time, it is a central element of organizing teamwork.

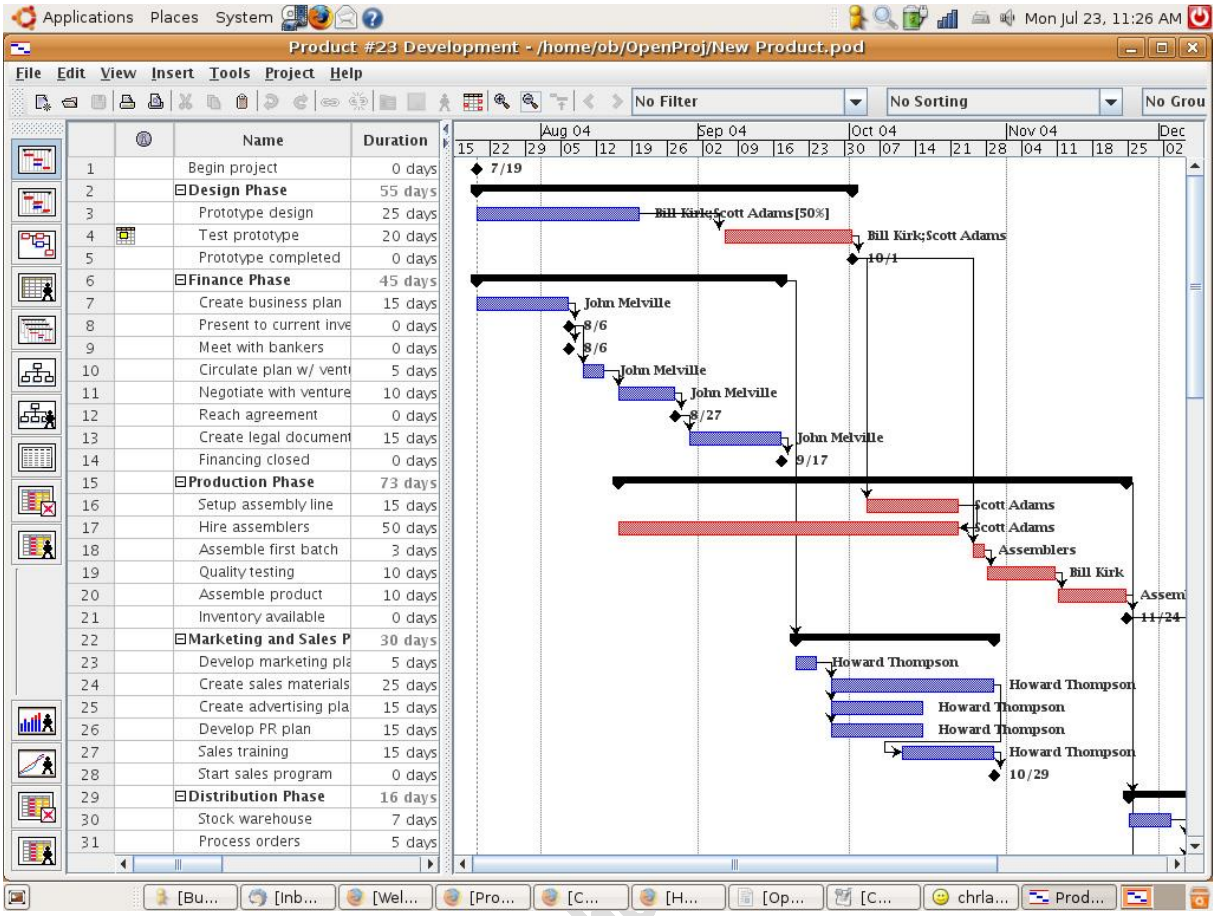
Using this kind of computer tools is far from being the privilege of businesspersons. In fact, projects from the simplest to the most complex ones may be administered more efficiently this way. In the field of humanities, project management software can facilitate organizing individual research as well as managing teamwork, for instance, in publication projects, exhibitions, workshops, congresses etc. (There are specialized applications too, sharing some aspects of general-purpose project management tools, designed for the specific tasks of publishing – especially periodicals – or conference organizing, widely utilized also in the humanities. In other respects, these applications may be also seen as specialized content management systems.)

iii) Key elements in project management software

The core element of a project management tool is a database containing information about every important aspect of the elements of a project:

- a project timeline (or schedule), i.e. a calendar with (at least) the start and end dates of the project
- tasks (and subtasks) to be completed in order to realize the objectives of the project
    - each task has specific values with regard to some essential parameters such as duration, deadline, cost, workload, etc. (for instance, how many hours of work is necessary for carrying out the activities required by a given task)
- necessary (human and material) resources, or available resources
    - each resource has specific values with regard to some essential parameters such as availability dates/periods/durations and costs (for instance, how many hours a project collaborator can spend daily on tasks assigned to him, and when)
- important events and dates (milestones having a decisive importance in the success or failure of the project)
- relations (e.g. tasks assigned to participants)
- dependencies (between tasks, etc.): temporal and/or logical ones (for instance, a task may necessarily have to be completed before the starting moment of another one, which depends on the result of the first one)
- participants (not necessarily identical to human resources mentioned above) with whom the project is shared in collaboration.

Project management software allow users to get an overall view of their project in the form of a Gantt chart. It is a comprehensive way of overseeing the timeline, tasks, dependencies etc. of the whole project. Ideally, a project management tool integrates well with other systems such as calendars or e-mail as it needs to be a collaboration utility among participants, allowing follow-up as well as reporting/evaluation at any stage of the project lifetime. In order to keep the project on schedule, managers and participants receive alerts and reminders on upcoming events such as activity deadlines.



		Name	Duration
1		Begin project	0 days
2		⊟Design Phase	55 days
3		Prototype design	25 days
4	▦	Test prototype	20 days
5		Prototype completed	0 days
6		⊟Finance Phase	45 days
7		Create business plan	15 days
8		Present to current inve	0 days
9		Meet with bankers	0 days
10		Circulate plan w/ vent	5 days
11		Negotiate with venture	10 days
12		Reach agreement	0 days
13		Create legal document	15 days
14		Financing closed	0 days
15		⊟Production Phase	73 days
16		Setup assembly line	15 days
17		Hire assemblers	50 days
18		Assemble first batch	3 days
19		Quality testing	10 days
20		Assemble product	10 days
21		Inventory available	0 days
22		⊟Marketing and Sales P	30 days
23		Develop marketing pla	5 days
24		Create sales materials	25 days
25		Create advertising pla	15 days
26		Develop PR plan	15 days
27		Sales training	15 days
28		Start sales program	0 days
29		⊟Distribution Phase	16 days
30		Stock warehouse	7 days
31		Process orders	5 days

Figure 2 A project overview with a Gantt chart

iv) Using project management software

Using a project management tool helps not only to follow efficiently the evolution of a project but also to make oneself a clear idea of what may be involved in/by it from the very beginning. Difficulties may reside less in technical skills required for handling the tool itself, which is usually not very demanding in this respect, than in concretizing rather general (and abstract) ideas of what we would like to achieve by the project. Here is a non-exhaustive list of points one has to think through when mounting a project.

- aims/objectives of the project
  - concrete (material) outcomes to be obtained (deliverables)
  - breakdown of main objectives into partial ones, work phases and tasks
  - relations and dependencies between particular work phases / tasks
- schedule of the project
  - starting and ending dates
  - critical points / dates (milestones)
  - time needed for each task / work phase (N.B. in most cases, it is not the “inherent”, “objective” timing of individual tasks that is

    added up to determine the overall duration of the project but, inversely, individual timings of tasks have to be calculated regressively from the amount of time at our disposal for the whole project, determined by external factors)

- resources
    - budget / costs
    - material resources (e. g. hardware equipment, software, books, services – like subscriptions or travel –, etc)
    - human resources
    - needed vs. available resources
    - availability of resources with respect to time
- possible risks and changes
    - multiple what-if scenarios in order to foresee possible project evolutions

Precise data concerning deliverables, resources / costs and schedule, agreed by everyone participating to the project, define a so-called baseline against which real progress can be constantly compared and evaluated. Evidently, it supposes that one should continuously record the facts relevant to the project within the software. We can then see how contingent changes affect the realization of the project and what modifications seem necessary to face the consequences. As a result, an updated baseline can be defined.

v) Some examples of project management tools

name of software	author / publisher / company; website	main features
Project	Microsoft; https://products.office.com/en-us/Project/	• market-leading commercial software (compatibility of project files made with other project management applications with MS Project is usually marked as a key feature) • sold in different versions according to customer profiles (team members, project managers or executives) and licensing/delivery model (standalone or cloud-based) • can collaborate with other MS Office apps • state-of-the-art project management utilities • allows to anticipate possible evolutions of projects

name of software	author / publisher / company; website	main features
ProjectLibre	Marc O'Brien, Laurent Chrettieneau; http://www.projectlibre.org/	• award-winning popular open source software intended to be a replacement for MS Project • free desktop application (with planned cloud extension) • compatibility with several versions of MS Project files
Gantter	InQuest Technologies, Inc.; http://www.gantter.com/	• free cloud-based project management tool • compatibility with MS Project • integrates with leading cloud storage providers • collaboration in the framework of Google Drive • three versions for different cloud-based context and a Chrome extension for offline work
Wrike	Wrike, Inc.; https://www.wrike.com/	• different versions, with a free one among them • optimized for tablets (and smartphones) as well • compatibility with MS Project • integrates with standard office and communication tools as well as cloud storage services

3) Managing references and bibliography

i) Fastidious tasks and smart solutions

Texts written by others play an important role in academic writing: not only has the author to situate his work in a given scientific context but data and claims of publications relevant to the topic of his research inevitably serve for him as a kind of staring point. If we just take a closer look at the bibliography of some books or papers, observing by the way how references are made in the flow of the text to the sources being listed, we may formulate the following observations:

- An important number of sources are referred to even in the case of a relatively short publication.

- Bibliographical data of different types of works are presented in a differentiated yet consistent way.
- In a given bibliography, records have an identical data structure (e. g., for a book, name of the author, year of publication, title, place of publication, name of the publisher, number of pages, etc.).
- Every data field is filled in uniformly (e. g. the given names of authors are either printed out in their full form or reduced to initials, and this applies to every record in a consequent way).
- A uniform typography is applied to data of the same type (e. g. publication years put between brackets, book titles printed in italics, etc.).
- In the main text of the publication, references to bibliography records strictly follow some basic patterns. They are also displayed with specific and constant typographical attributes (e. g. an author's name in small caps followed by the publication date of his work being cited as well as a page number separated by a colon and the whole put between brackets).
- There is a bi-univocal correspondence between references in text and bibliography records, i.e. a bibliography item corresponds to each publication having been referred to in the main text, and nothing figures in the bibliography unless it was explicitly mentioned in the main text itself. Similar publications are formally distinguished (e. g. publications of the same author from the same year).
- Several models (styles or norms) of presenting references and bibliography do exist and journals, reviews or book series adopt one or another among them. In no case should we find different referencing norms followed in two publications that belong to the same editorial framework. These models or styles usually have some kind of broadly known name or identifier (e.g. Chicago).

Considering all these requirements, one easily understands that dealing with references is one of the heaviest burdens in academic writing. Still, it is all only about mere data management and not about the essential, creative part of the endeavour. At the same time, it is quite obvious that we have to address there rather mechanical, thus, automatable tasks. As an intelligent solution, bibliography management tools come into sight. These tools are known by the name of personal bibliographic management software or reference or citation management software.

Computerized referencing is a standard procedure in many scientific fields, especially in life and other "hard" sciences. Indeed, there are no theoretical or practical limitations to use it in any discipline; however, it still seems less adopted in arts and humanities (at least in Hungary). Nevertheless, even if not every potential source text is available in an electronic, online accessible form, bibliographical data of almost every publication, printed and/or e-published, is obtainable from the Internet, most of the times in one or more normalized formats. (Examples of such formats are BibTex, EndNote, MARC or RIS.) Online bibliographical and citation databases like Scopus or Web of Science have become indispensable research tools. In order to take full advantage of such resources, we need citation management software. Although their features are very similar, one should pay attention to choose carefully the reference tool fitting the best one's needs – and one's budget. In fact, market-leading bibliography management services are very expensive. Still, academic institutions usually have corporate subscriptions to one or more of these services and you may benefit from it as a member. Therefore, the first step of your search for the ideal tool should be a visit on the website of your institution's library.

ii) Functionalities and use of reference / bibliography management tools

If you are just about to write a dissertation as a part of your curriculum at the university, it is maybe more simple and efficient to resolve problems of referencing manually, given that even the simplest software has a learning curve. Moreover, feeding a personal bibliographical database is a time-consuming activity, which is getting more profitable more you are making references to the items it contains. However, if you have a longer work to write or if you are looking forward to a followed activity in academic writing, you will have to handle an increasing number of bibliography items, with many among them that you should cite all the time in several publications and that considerably increase the repetitiveness of those mechanical editing tasks. Therefore, a smart way of dealing with this part of the work consists in using some bibliography management software for automating its fastidious steps. Actually, this is the only way you can securely meet all the abovementioned consistency criteria.

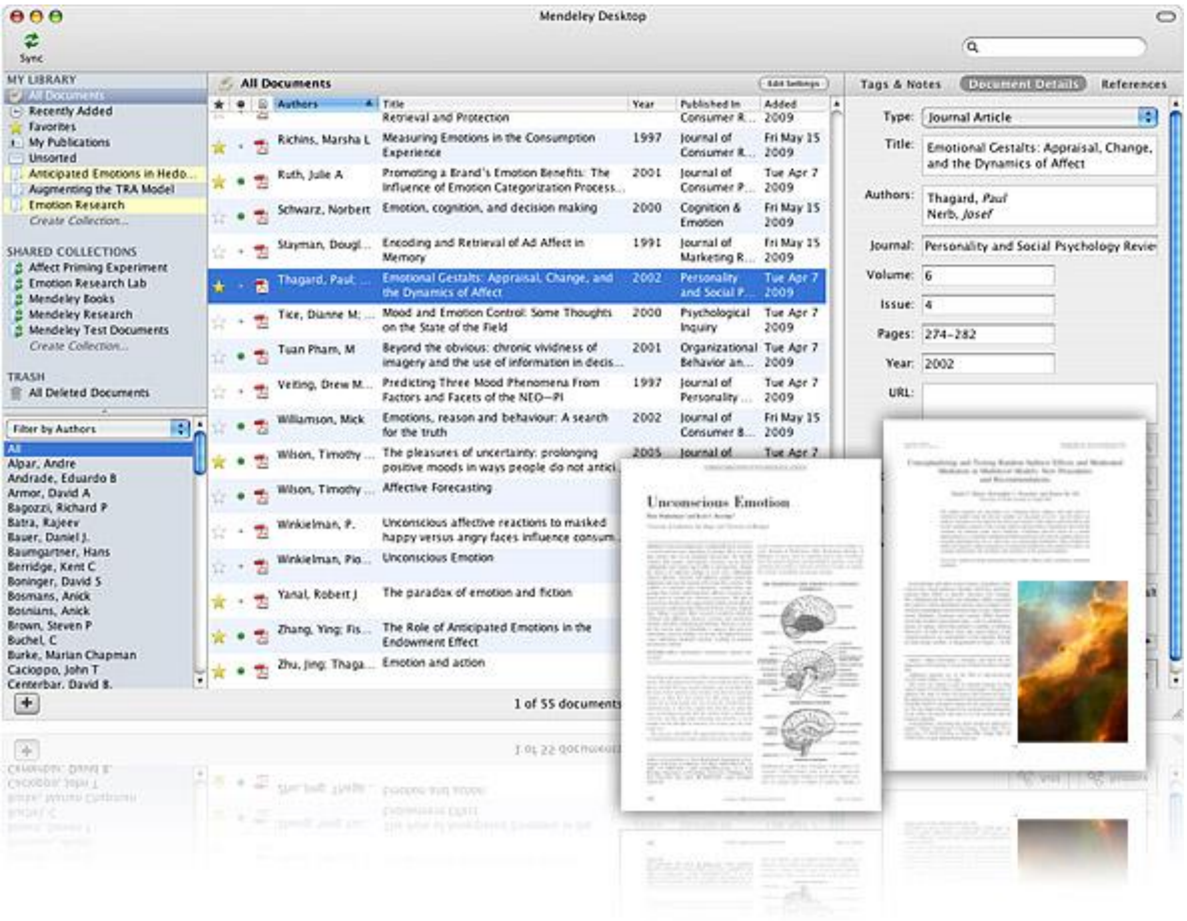
Working with tools of this kind implies three types of activities:

1. building a personal bibliography database
2. inserting references (and, contingently, citations) in what you are writing
3. generating the bibliography at the end of your paper

The first one is a continuous activity as long as you do research and work on your paper(s). The more you feed your bibliography database, the more your software will facilitate your writing. Following the tool(s) at your disposal as well as the sources you need to refer to, there are various methods of database building. If your referencing system is combined with a large online database, a simple search within the system can provide you with the necessary data. Even online full-text version of publications may be available, which is a very convenient way of dealing with sources. In other cases, you may have to search elsewhere for bibliography items you need and import records in your bibliography management tool. Sometimes, it may be necessary for you to enter manually bibliographical data of some sources by filling in a form provided by the system. It may also be possible to upload and store full-text copies of sources (usually in PDF format). Some software enable you to add annotations as well to your stored texts, or even to carry out planning and drafting operations, becoming this way veritable academic writing assistants.

When working on your paper, instead of typing in references, you insert them in the text by retrieving your bibliography records and choosing the necessary ones in a dialogue box of your word processor. That is to say, reference management software must integrate with document editing tools, so they are usually distributed together with word processor plug-ins. (N.B.: document editors usually have built-in bibliography building and referencing functions, too.) Users can choose the reference format they have to adopt, and references will be automatically inserted in the required form. If some special format is not provided in the default set, you may adjust the settings according to your (or your editor's) preferences. You can also fine-tune a given reference for fitting it in a particular context. It is worth noting that *references are not added as simple text but as special fields*, which is another great advantage compared to traditional referencing method. This way, *you can update your bibliography entries or change reference styles at any time: all changes will be automatically replicated on every token*.

The last step is perhaps the simplest one. In order to compile the list of bibliographic sources you have used and mentioned in your work, you simply need to put the cursor at the point of your document where the bibliography must be placed, and give your software the instruction of generating it automatically.



Authors	Title	Year	Published In	Added
	Retrieval and Protection		Consumer R...	2009
Richins, Marsha L	Measuring Emotions in the Consumption Experience	1997	Journal of Consumer R...	Fri May 15 2009
Ruth, Julie A	Promoting a Brand's Emotion Benefits: The Influence of Emotion Categorization Process...	2001	Journal of Consumer P...	Tue Apr 7 2009
Schwarz, Norbert	Emotion, cognition, and decision making	2000	Cognition & Emotion	Fri May 15 2009
Stayman, Dougl...	Encoding and Retrieval of Ad Affect in Memory	1991	Journal of Marketing R...	Fri May 15 2009
Thagard, Paul; ...	Emotional Gestalts: Appraisal, Change, and the Dynamics of Affect	2002	Personality and Social P...	Tue Apr 7 2009
Tice, Dianne M; ...	Mood and Emotion Control: Some Thoughts on the State of the Field	2000	Psychological Inquiry	Tue Apr 7 2009
Tuan Pham, M	Beyond the obvious: chronic vividness of imagery and the use of information in decis...	2001	Organizational Behavior an...	Tue Apr 7 2009
Veiting, Drew M...	Predicting Three Mood Phenomena From Factors and Facets of the NEO—PI	1997	Journal of Personality ...	Tue Apr 7 2009
Williamson, Mick	Emotions, reason and behaviour: A search for the truth	2002	Journal of Consumer B...	Fri May 15 2009
Wilson, Timothy ...	The pleasures of uncertainty: prolonging positive moods in ways people do not antici...	2005	Journal of	Tue Apr 7
Wilson, Timothy ...	Affective Forecasting			
Winkielman, P.	Unconscious affective reactions to masked happy versus angry faces influence consum...			
Winkielman, Pio...	Unconscious Emotion			
Yanal, Robert J	The paradox of emotion and fiction			
Zhang, Ying; Fis...	The Role of Anticipated Emotions in the Endowment Effect			
Zhu, Jing; Thaga...	Emotion and action			

1 of 55 documents

Type: Journal Article

Title: Emotional Gestalts: Appraisal, Change, and the Dynamics of Affect

Authors: Thagard, *Paul*
Nerb, *Josef*

Journal: Personality and Social Psychology Revie

Volume: 6

Issue: 4

Pages: 274–282

Year: 2002

URL:

Figure 3 Interface of a bibliography manager (Mendeley)

iii) Some examples of bibliography/citation management tools

N.B.: Docear, mentioned in section 1)a)iv) as a special variety of mind mapping tools, belongs to this type of software as well. – You may find, of course, complete and detailed lists and comparisons of citation management tools on the Internet. In the bibliography section below, we have included two instructive posts from the Docear website (Beel 2013 and 2014) in order to provide the reader with some insightful analyses of what should be taken into account when choosing a reference management tool. Data contained in the table below illustrate only main types and orientations of these tools, and focus on some global characteristics more than particular functionalities.

tool	author / publisher / company; website	main features
EndNote	Thomson Reuters; http://endnote.com/, www.myendnoteweb.com	• one of market-leading commercial reference management software • desktop, mobile and online versions • EndNote basic: free online version • institutional subscriptions • integrates with MS Word • advanced functionalities for organizing research, publishing and sharing as well
JabRef	JabRef Development Team (Morten O. Alver, Nizar N. Batada, et al.); http://jabref.sourceforge.net/	• open source desktop application • uses a widespread but more specific bibliography format (BibTeX, developed for the LaTeX document preparation system) • importing, editing and exporting functions for bibliography records • connects to several important external databases • search and classification functions • integrates with external applications (web browser, PDF viewer, some document editors) • use with MS Word is possible but not as straightforward as with some other tools

tool	author / publisher / company; website	main features
Mendeley	Elsevier B.V.; www.mendeley.com	• one of the most popular tools (originally independent but later acquired by one of the market-leading scientific publishing companies) • free reference manager and academic social network • commercial premium and institutional packages • has mobile apps as well • advanced features for work with full-text documents, annotation and collaboration • compatible with MS Word, LibreOffice and BibTeX
RefWorks, Flow	ProQuest; www.refworks.com ; https://flow.proquest.com/	• two different products of a market-leading information and technology solutions provider • both integrate with MS Word • RefWorks: ○ online research management, writing and collaboration tool ○ commercial (institutional or individual subscriptions) with free trial ○ focuses only on referencing and citing • Flow: ○ free for individuals (with limitations) ○ available to institutions and companies ○ extended functionalities compared to RefWorks ○ supports reading, annotating, and collaborating as well ○ focuses on full-text documents and collaboration

tool	author / publisher / company; website	main features
Zotero	Roy Rosenzweig Center for History and New Media at George Mason University, Corporation for Digital Scholarship; www.zotero.org/	• free software • registering to Zotero File Storage is free as well and gives access to cloud-based services (larger storage quota being available for subscription) • supports collecting, organizing, analyzing and sharing research data • interacts with the user's web browser for collecting any content and allows to import documents for building one's personal library • allows adding notes to records • integrates with MS Word and Open Office via plugins

Further reading, sources/resources

Beel, Joeran (2013) *What makes a bad reference manager?* Docear, http://www.docear.org/2013/10/14/what-makes-a-bad-reference-manager/

Beel, Joeran (2014) *Comprehensive Comparison of Reference Managers: Mendeley vs. Zotero vs. Docear*, Docear, http://www.docear.org/2014/01/15/comprehensive-comparison-of-reference-managers-mendeley-vs-zotero-vs-docear/#summary

Beel, Joeran and Gipp, Bela and Langer, Stefan and Genzmehr, Marcel (2011) "Docear: An academic literature suite for searching, organizing and creating academic literature" in *Proceedings of the 11th ACM/IEEE Joint Conference on Digital Libraries (JCDL'11)*, Ottawa, Ontario, Canada, pp 465-466.

Buzan, Tony (2006) *Mind Mapping*. Pearson Education

Davies, Martin (2011) "Concept mapping, mind mapping and argument mapping: what are the differences and do they matter?" in *Higher Education*, Volume 62, Issue 3, pp 279-301 – http://media.usm.maine.edu/~lenny/critical%20thinking%20and%20mapping/mind%20mapping.pdf

Fenner, Martin (2010) "Reference Management meets Web 2.0" in *Cellular Therapy and Transplantation*, Vol. 2, No. 6, pp 1-3 – http://www.h2mw.eu/redactionmedicale/2010/10/Ref%20management%20%26%20eb%202.0_CTT-2-6-2010-Fenner_en%5B1%5D.pdf

Hensley, Merinda Kaye (2011) "Citation management software: Features and futures". *Reference & User Services Quarterly* 50, no. 3, pp 204-208. – https://www.ideals.illinois.edu/bitstream/handle/2142/18659/RUSQ_Fall2010final_Hensley.pdf?sequence=2

Klastorin, Ted (2010) *Project Management: Tools and Trade-Offs*. Pearson Learning Solutions

Creating and Managing Databases

István Szekrényes

1) General presentation

Data collection, digitalisation, creation of media and annotation files are very beneficial and impressive works in humanities, but they do not support properly the final research goals if there is no systematic order in the resulted corpus and we could not process the collected data in an effective, computer-assisted way. In a database, data have to be stored in a structured form to make the collection manageable. This "structured storage" has a conceptual and a technical aspect. At first, a logical data model (various types are available: hierarchical, relational, object oriented etc.) is needed, which can describe the data: the type of entities with their properties and relations. After this conceptual planning, a suitable database management system (DBMS) has to be used for the technical implementation of the database. The standard, three-level architecture of databases can be also divided to the physical, the conceptual and the external (results of queries) levels.

In everyday life, we can come in contact with various types of databases. They also serve various types of purposes, and they are required everywhere where high amounts of data have to be managed in a fast and reliable way. Companies, institutions, public services often use a database management systems to make their work easier, for instance, in managing orders, customers, students etc.

In humanities, databases can simply serve administrative tasks like an electronic catalogue system in a library, but they are very useful for direct research as well. For instance, database queries are suitable for performing various kinds of quantitative analysis or filtering the research material.

i) Main functionalities of the software type

As it was mentioned above, the most important function of database management systems is the structured storage of data, but they have to fit other requirements as well:

- The possibility of entering, modifying and querying the data
- Ensuring parallel transactions
- Providing different user accounts and privileges
- Redundancy-free storage (duplicates are not allowed in a database)
- Securing the integrity of the data (contradictions are not allowed in a database)
- The independence of the program and the data
- Data security (by logging pre-conditions etc.)
- Remote (client-server) access to the database
- Privacy (password-protected user accounts)

The most of the database management systems support the relational data model and use the SQL (Structured Query Language) programming language, which is specialized for managing data in a relational database. SQL is based on the relational algebra, and consists of a data definition and a data manipulation language. Its implementations can also use graphical user interfaces to make the work more user-friendly. Web sites can also communicate with SQL database servers via PHP[1] commands, therefore users and admins often use an online interface to perform database queries or other operations.

ii) Particular software of the given type

Commercial software

The most well-known commercial database management programs are published by Microsoft and Oracle companies: Microsoft Access and Oracle Database. Both of them use the SQL language and implement the client-server architecture.

Free/open source software

The most popular free[2] and open source database management software is MySQL, which also uses the SQL language and supports the client-server architecture. One can use the program in command line mode or with a graphical user interface (MySQL Workbench Utility, phpMyAdmin).

MySQL:

author/publisher/company:	Oracle
website:	http://www.mysql.com/
principal characteristics:	open source, client-server architecture, multi-user, multi-threaded, using SQL language and relational data model

2) Using relational databases

Before using any instance of the above listed database management software, it is recommended to know how a relation database works in general. The essence of the relational data model is constituted by the *relational schemas* and their relationship with each other, where every scheme represents an entity type with its

[1] PHP is a server-side programing language designed for web development. One can find more information on the official website of the project: https://php.net/.

[2] Commercial versions (MySQL Enterprise Edition, MySQL Cluster CGE) of the software are also available with extended features on the official website.

attributes (common properties). For instance, in case of a library catalogue system, the *volume* (books, journals etc.) is a possible entity type with the attributes: *call number*, *type*, *author*, *title*, *publisher*, *release* etc. The relational schema of this entity can be represented in the following form:

```
VOLUME (ISBN, TYPE, AUTHOR, TITLE, PUBLISHER, RELEASE)
```

In practice, these abstract schemas are implemented as *relational tables* containing those entities (e.g. the “volumes” in the above example) which can be described by the attributes of the schema. Each *record* of the relational table has these attributes with a concrete value. If data are inserted into the table, the result can be imagined as a Microsoft Excel spreadsheet (see Table 1), where the names of the columns are the *attributes*, the lines are the entities (*records*), and every cell contains an *attribute value* related to a certain attribute of an entity.

Table 1: Data in a relation table

CALL NUMBER	TYPE	AUTHOR	TITLE	PUBLISHER	RELEASE
99921-58-10-7	book	Kneale, W	Introduction to…	Springer	2001
9971-5-0210-0	book	Johns, A	The essence of…	Oxford University Press	1996
960-425-059-0	book	Austin, B	Language and…	Oxford University Press	2012

The main difference between a simple spreadsheet and a relational table is an insured constraint that every record (every entity) has to be individual, duplicates are not allowed. In databases, an *artificial id* is generally used to ensure this clarification. In Table 1, the “call number” is this special attribute, the *primary key* of the relational table, which is always individual. The other attributes are not surely suitable to distinguish the records from each other. For instance, it is easily possible that two books are presented with the very same bibliographical data in the same library, and the only difference will be the call number.

The relational databases generally contain more than one relational table, and they are in connection with each other. For instance, in a speech database, one table can contain the data of the speakers and another one the data of the recordings, therefore there are two relational schemas in the database:

```
SPEAKER (ID, GENDER, AGE)

RECORDING (ID, DATE, CONTENT, SPEAKER_ID)
```

The reason of using two separate tables is the redundancy-free data storage. If one speaker is represented in more than one recording and every data would be in the same table (either related to the speakers or the recordings), the same speaker's data should be also represented in every related record like in Table 2.

Table 2: Redundant data storage

RECORDING_ID	DATE	CONTENT	SPEAKER_ID	GENDER	AGE
1	2014.06.01	reading	1	male	25
2	2014.06.01	singing	1	male	25
3	2014.06.02	reading	2	female	23
4	2014.06.02	singing	2	female	23

Instead of this redundant data storage, two separate tables can be used which are connected through the *id* of the speakers:

Table 3: Relational table of speakers

ID	GENDER	Age
1	male	25
2	female	23

Table 4: Relational table of recordings

ID	DATE	CONTENT	SPEAKER_ID
1	2014.06.01	reading	1
2	2014.06.01	singing	1
3	2014.06.02	reading	2
4	2014.06.02	singing	2

The "speaker_id" attribute in Table 4 is a so-called *foreign key* (or *external key*) that refers to the *id* attribute of Table 3. In this way, different relational tables can be connected in a relational schema.

In Figure 2, an EER diagram (enhanced entity–relationship model) of a relational database can be seen, and it is created by the MySQL Workbench utility. There are four different relational tables in the schema: *interview*, *recording*, *interviewer* and *interviewee*. Their relationships are also represented in the diagram. For instance, there are three foreign keys (*recording*, *interviewer*, *interviewee*) in the interview table, and each of them refers to the primary key of a particular table (marked by a yellow key). The records in the interview table only contain the text of the interview, but through these foreign keys they also refer to the data of recording (*date*, *file-size*, *duration*) and the data of participants (*gender*, *age* and *provenance*). The *domains* of the attributes can also be seen in the diagram. They are to limit the possible values of the attributes. For instance, the *id* attribute of the interviewer table has only integer (INT) values while the *gender* can contain only strings (VARCHAR) with maximum six characters.

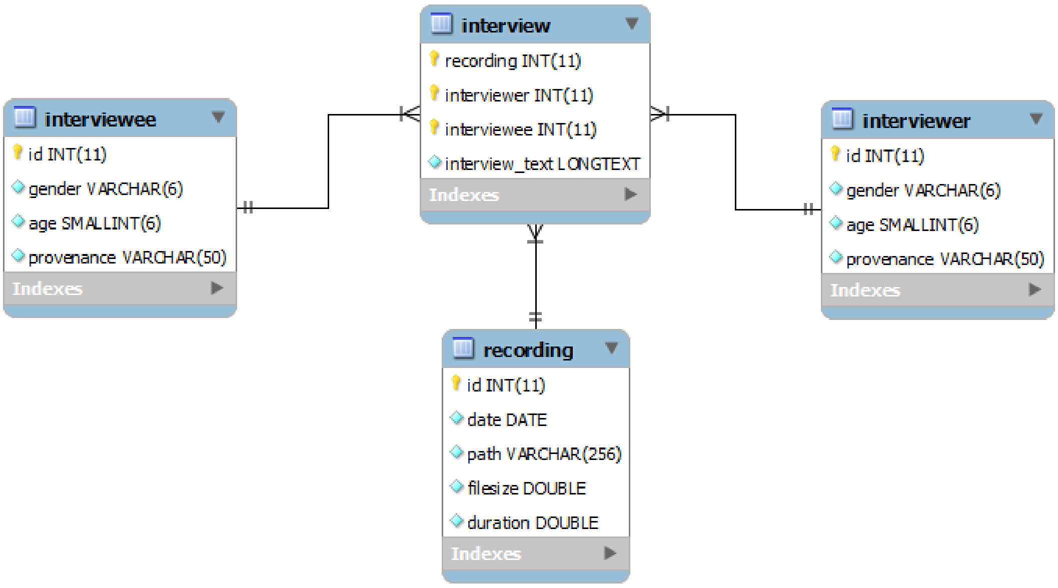


Figure 2: EER diagram of a relational database

Two relational tables could have three different relationships (1:1, 1:N, N:M) depending on how many relation can be established by one entity from the first and the second tables. For instance, the *interviewer* and the *interview* table (In Figure 2) has a 1:N relation because only one interviewer can participate in an interview, but an interviewer can participate more than one interview.

i) Creating relational databases in MySQL

After planning a particular relational schema, standard SQL statements or an available graphical interface can be used to create the database itself. First the MySQL client has to be connected to the local or a remote MySQL server. The local one can be accessed under a special hostname (named *local host*):

```
mysql -h localhost -u myUserName -p myPassword myDataBase
```

The following SQL statements (separated by semicolons) will create the "interview" database which was represented in Figure 2:

```
CREATE SCHEMA interviews;

USE interviews;
```

```
CREATE TABLE interviews.recording (

id INT NOT NULL,

date DATE NOT NULL,

path VARCHAR(256) NOT NULL,

filesize REAL NOT NULL,

duration REAL NOT NULL,

PRIMARY KEY (id));


CREATE TABLE interviews.interviewer (

id INT NOT NULL,

gender VARCHAR(6) NOT NULL,

age SMALLINT NOT NULL,

provenance VARCHAR(50) NOT NULL,

PRIMARY KEY (id));

CREATE TABLE interviews.interviewee (

id INT NOT NULL,

gender VARCHAR(6) NOT NULL,

age SMALLINT NOT NULL,

provenance VARCHAR(50) NOT NULL,

PRIMARY KEY (id));


CREATE TABLE interviews.interview (

recording INT NOT NULL,

interviewer INT NOT NULL,

interviewee INT NOT NULL,

interview_text LONGTEXT NOT NULL,

PRIMARY KEY (recording, interviewer, interviewee),

FOREIGN KEY (recording) REFERENCES recording(id),
FOREIGN KEY (interviewer) REFERENCES interviewer(id),
FOREIGN KEY (interviewee) REFERENCES interviewee(id));
```

The created tables will contain empty columns, and the domains will determine what kind of data (textual, numeric etc.) can be inserted into a certain column. New records can be inserted with the INSERT statement:

```
INSERT INTO interviewer VALUES (1, 'female', 34, 'London');
```

As it can be seen from the above example, the attribute values have to be inserted in the same order as the attributes are in the schema.

ii) Using SQL queries

SQL queries can be performed with the SELECT statement which generates a new, resulting table based on the selected ones in the schema. The users can be in direct contact only with these external views of the database (the external level of the three-level architecture).

If column names and conditions are not specified, the SELECT statement selects every record from a relational table. For instance, every data will be queried from the volumes table in a library catalogue system with the following use of SELECT statement:

```
SELECT * FROM volumes
```

It is possible to limit the number of the columns in the resulting table if the names of the columns are specified:

```
SELECT author, title FROM volumes
```

Conditions can also be added to the query. In this case, only those records are inserted into the resulting table which satisfy the condition. The conditions can contain regular expressions, different kinds of logical (AND, OR etc.) and other operators. The following SQL statement will list only the books with a certain author from the table of volumes:

```
SELECT * FROM volumes WHERE type='book' AND author='Johns'
```

The resulting tables can be sorted with the ORDER BY expression using a certain column to order the data in a numerical or alphabetical order. The data will be organized on the basis of the values of the *title* attribute in the following example:

```
SELECT * FROM volumes ORDER BY title
```

Arithmetical operations can also be included in the query. For instance, the COUNT function returns the number of records in the resulted column. The output of the following SQL statement will be the number of books (the records with 'book' type) in the *volumes* table:

```
SELECT COUNT(type) FROM volumes

WHERE type = ‘book’
```

Different relational tables can be joined with the JOIN expression using the foreign keys as the basis of connection. In the following example, the *interviewer* and the *interview* tables (from Figure 2) will be joined based on the *id* attribute of the interviewers:

```
SELECT interviewer.id AS interviewer_id, COUNT(interview.id) AS
number_of_interviews

FROM interviewer JOIN interview

ON interviewer.id = interview.interviewer

GROUP BY interviewer.id
```

The above statement also contains the GROUP BY expression which always has to be used in conjunction with an aggregate function (AVG, COUNT etc.) to group the result-set by one or more columns. In the example, the resulting table will have two columns: the id of the interviewers and the number of the interviews which were guided by them:

interviewer_id	number of interviews
1	24
2	231
3	56

iii) Graphical and online interfaces

In the above examples, database operations were described in the SQL language but these command line methods are not the only way of managing them. For instance, in MySQL Workbench Utility, a complex menu system, dialog boxes and a diagram designer tool can be used for data definition, data manipulation and database server management. One can try the web interface of phpMyAdmin, which is a PHP framework using HTML[3] forms for the same operations. However, these tools can make the work much easier, they all have a program-specific usage, which cannot be detailed in this book. More information can be found in the official manuals.

[3] HyperText Markup Language (HTML) is standard *markup* language for describing the structure, the content and the style of web documents. HTML codes can be translated and displayed as web page by any web browser.

Further reading, sources/resources

In this chapter, we could not discuss all the available features and functions of the SQL language and database management. Further reading is recommended if one wants to seriously deal with the above topics. MySQL has a very good and complex documentation on the official website[4]. There is another online source, named w3shools.com, which also has a not so detailed but very useful section[5] for learning SQL. The latter one is highly recommended for beginners because there are many examples, and users can test SQL commands on a real, learning database which can be accessed from the site.

[4] http://dev.mysql.com/doc/

[5] http://www.w3schools.com/sql/

Multimodal Data: from Communication to Annotation (and Vice Versa)

András Földesi

This chapter is organized along frequently asked questions by the present and potential users of the HuComTech corpus[1] so as to facilitate human-machine communication. In order to exchange information with nonhuman (technical) beings, i. e. computers, as a first step we have to make the communication of machines with humans easier. How can you get into contact with a computer in case you would like to get (see or hear) a correct answer from a machine? The human part of such a conversation expects more than just a response, that is, more than a reaction to a stimulus. The human user is waiting for an answer that matches his or her expectations regarding the situation, the context, even his or her mood. Making human behaviour interpretable for machines is a very complicated, almost unattainable task. Its implementation can only start on a very basic level. You may think that a mimetic copy of human social behaviour might suffice at the very first stage. Below we are going to provide an insight into a detailed method of description as well as into the ways in which natural language and nonverbal behaviour can be represented with annotations, which can be processed by machines.

a) On the process of annotation

Annotation means segmenting a corpus into units and labelling/tagging these units with respect to predefined categories (e. g. parts of speech, syntax, speech acts, etc.). Our working group in the HuComTech project has been watching and listening to the videos of the recorded dialogues of the HuComTech corpus[2]. The dialogues were more or less guided and controlled (formal simulated job-interviews on the one hand, and informal interviews on the other). The annotators, who were responsible for describing and interpreting each and every physical moment and contextual information within the dialogues, had **to segment** these conversations into minimal meaningful physical and interpretative events, which, in turn, had to be **assigned tags** (formal or functional descriptions) that can be made machine-readable (Hunyadi et al. 2012).

[1] HuComTech website: https://hucomtech.unideb.hu/hucomtech/

[2] The HuComTech corpus is described in detail in Hunyadi, L., Földesi, A., Szekrényes, I., Staudt, A, Kiss, H., Abuczki, A. & Bódog A. (2012): Az ember-gép kommunikáció elméleti-technológiai modellje és nyelvtechnológiai vonatkozásai. In: Kenesei, I., Prószéky, G. & Várady T. (Eds.): *Általános Nyelvészeti Tanulmányok XXIV. Nyelvtechnológiai kutatások* (pp. 265–309). Budapest: Akadémiai Kiadó.

Annotators were aware that the understanding and interpretation of different kinds of human behaviour is rather subjective. The subjectivity of judgment regarding the events that one witnesses is the first question one usually raises. What is more, subjectivity is doubled when we consider that annotation is annotator-dependent. You may feel uncertain about which gestures or series of gestures, which tone of speech can mean the same thing during a conversation. Or cannot it vary across time even in the case of the same person?

Above all, we need to identify and generalize certain recurrent minimal events and series of such events in a dialogue so as to capture their **objective, speaker-independent nature**. A person-dependent function of a tiny movement or spoken sound can be attributed a function shared by most or all of the interviewed speakers. When we see some body movement (of the head, hands, etc.) we need to find out if this movement can be attributed a communicative function, whether it can be considered as a sign of some behavioural importance. Roughly, **a sign is** usually defined as **something that stands for something else.** Whereas it would be the best for a machine to match a single physical sign with a single behavioural function, human behaviour is far from one-to-one matching. When we encounter, for instance, a moment of a person's lowering of his/her head, are we sure that it is intentional and it has any communicative meaning at the given moment? In this case we have several choices: ‘no meaning’, ‘agreement’, ‘turn-ending’, ‘uncertainty’, etc. Fortunately for us, humans, but unfortunately for the machine, such grains of behaviour cannot be analyzed just alone, in themselves, independently of other markers of multimodal events.

What we see at this point is that while we are trying to understand and interpret bits and pieces of human behaviour, we are doing information processing. **Processing involves encoding and decoding information.** We encode the information associated with our behaviour, then, in order to transfer the intended (or assumed to be intended) message, we decode this information.

Communication is generally multimodal. It means that information is conveyed through various channels, both visual and auditory, by verbal and nonverbal means (Massaro 1987, Kendon 2004); by words and the way these words are pronounced (prosody), and by gestures (body language).

In order to cover all these aspects of communication between people, various kinds of annotations were developed for the HuComTech corpus, each of them corresponding to a given modality.

b) Data analysis

i) Audio annotation

One of the most common annotation types is audio annotation. Based on the audio signal alone, one can transcribe the spoken utterances on a separate tier (level) for each of the speakers (in case of dialogues: 2). Since it often happens in spoken dialogues that the speakers simultaneously speak at the same time (overlapping speech), it is not quite straightforward to use an automatic STT (speech-to-text) analyzer (transcriber) for this purpose. In the HuComTech corpus, transcription was done by using the *Praat* phonetic analysis program allowing to carry out more sophisticated analyses (such as pause detection) as well. By doing so, continuous speech was segmented into smaller chunks, and these chunks were segmented by marking clause boundaries, disfluencies and pauses, events of turn-taking and turn-giving, etc.). As their boundaries were determined, they were assigned **timestamps** with a beginning and an ending time for each chunk. Since communication takes place in and across time, in order to determine the multimodal nature of communication, these timestamps are used to specify what markers of what modalities align (coincide) at a given point of time as well as across longer stretches of time. Since noise is an inherent part of conversation, it is also necessary to be annotated. Examples for marking such non-speech signals: {[]} = beep sound; {s} = sneezing, {l} = laughing.

It is also possible to identify the expression of emotions on the basis of audio only. The HuComTech corpus includes the annotation of emotions on the audio channel allowing to compare their attributes and frequencies to those of the – probably more obvious – video channel-based emotion annotation. What we find in general is that there appear a larger number of emotions identified in video than in audio only. Even though one can express some emotions by changing one's tone, loudness or speech rate, facial expressions are more effective and more decisive in this sense.

ii) Video annotation

By observing the video part of a recording, one can do two main kinds of annotation: that of the physical parameters, and that of interpretative values associated with these physical parameters. Certain physical parameters are more or less automatically recognisable: the identification and movement of the hand(s), the face and its parts (especially the eyes, nose and mouth) as well as the position of the head can quite reliably be done using automatic digital image processing. What remains essentially manual is the assignment of these observed positions, movements and trajectories to communicative functions. In order to do this, one needs more than just the description of physical parameters: one needs to study their temporal and sequential alignment and their specification as markers of various communicative functions. Automatic digital image processing is just the first step, the judgement of the human annotator cannot be replaced (at least at present).

During video annotation, performed using the Qannot software, the material is processed **frame by frame** (which meant units of about 270 milliseconds in the case of the HuComTech corpus). This was the minimal time period within which hand movements and head movements, the posture of the upper part of body, eye movements including blinks, etc. were observed, described and contextually interpreted. One can imagine how tedious it is to do such an annotation (with utmost responsibility, of course) throughout a recording of 20 minutes. While annotating head, hand and eye movements, posture change, the direction and repetition of such events was also to be considered (e. g.: right, left, up, down, etc.). The procedure of annotation remains the same as in the case of audio annotation: the series of events identified on the screen should be mapped onto an appropriate series of labels on different annotation tiers.

When video annotation is completed, what remains is the transformation of these labels into actions by the machine. Each of these highly detailed labels corresponds to a single operation in engineering terms: each movement of a single given machine part (such as one representing a joint of the arm) is determined by the actual parameters of a single given marker determined by the annotation label. The machine, in its turn, can be enabled to sense such single movements (and a complex of similar ones) and, finally, "interpret" them as our data suggests.

iii) Unimodal annotation

It is generally accepted in the literature that face-to-face communication is essentially multimodal, i.e. it requires both seeing and hearing (video and audio) in terms of percpetion, too. By seeing and hearing at the same time, by experiencing a multitude of modalities, one can come to a conclusion on what the given interaction is actually about. We do this holistically, i.e. without consciously separating the modalities and their markers and observing them individually before making some synthesis out of them. The machine, however, needs to know the exact state of each modality, each marker at a given time, and its current communicative state can only be composed of these single markers. Therefore, for human-machine interaction, it appears important to separately handle the two different approaches and modalities: unimodal and multimodal, and make final synthesis at the level of interpretation.

When performing the unimodal annotation of the video signal, the annotator cannot hear the sound channel and still, even without actually knowing the verbal content of the interaction, he or she has to (and mostly is able to) assign the given segments of conversation such labels as **turn-giving**, **turn-taking** and **turn-keeping**, labels representing **who is speaking at the moment and/or taking over the lead of the conversation.**

Amazingly, we can decide even without sound, who is the more active participant in the interaction. The labels *start speaking, intending to start speaking* and *end speaking*, refer to communicative functions that are detectable even without knowing the lexical content of the conversation. Also, although one does not know the actual content, one can see the speaker’s attitude; whether or how much he or she agrees or disagrees, whether he or she accepts what has been said; and how sure he or she is about what has been said.

It can also be observed how balanced the interaction between the two parties is. In addition to turn-take and turn-give, one can also observe *turn-keep* (when the speaker wants to keep the actual turn with him/her) or even *turn-grab* (breaking in to take the turn). Similarly, one can identify the degree of attention involved in the conversation just based on watching the video: *paying attention, calling attention* and *uninterested*. However, these two attributes are not always easy to differentiate due to the lack of supporting video: when somebody is looking at you strongly, staring at your face, it can be evaluated in two ways: 1. 'this person is strongly listening to me', 2. 'this is a provocative manner that appears to trigger my attention'.

Agreement has also been annotated in a unimodal (video only) annotation environment. On the top of the tiers subordinated to *AgreementClass* you can see *positive (+) agreement*, *negative (-) agreement*, and both are subcategorised into further tiers: on the positive side to *default* case of agreement, *full* agreement, *partial* agreement and *uncertain.* On the negative side, *agreement* labels include *default*, *block*ing and *uninterested*ness (being bored). A full agreement or a block disagreement are expressed intensively enough to be labelled, but default cases of both types can be disputable, for, we reserved the label *uncertain* for cases not falling into any of the categories.

The unimodal annotation contains a tier for marking the moments of the introduction of new information (with respect to what has been said previously in the context of the conversation). This is also an attribute that, probably somewhat paradoxically, can be inferred from patterns of behaviour without any knowledge about the content of the dialogue and, consequently, that of the new piece of information.

An example of the challenge of unimodal annotation, nodding, in our culture, is prototypically considered to be a marker of agreement while shaking one’s head that of disagreement. This is the principle the annotator would obviously follow in the case of unimodal annotation. However, it would be quite misleading, and multimodality would have to be taken into account in order to decipher the actual meaning of a head movement in either direction. Still, for the sake of proper machine implementations, we do need unimodality, even though the actual machine representation of several communicative functions (including agreement/disagreement) requires the multimodal complex of such unimodal descriptions.

iv) Multimodal pragmatic annotation

It has already been mentioned above that we need to slice up a conversation into marked chunks of events. When participating in a conversation, we do not only have information about discrete physical components of a conversation as described in terms of single visual and auditory markers of gestures and speech, but we interpret the complex of events as a whole. Accordingly, our annotation has to be extended to capture the interpretation of this complex in a holistic way, i.e. we also need information about the performed communicative acts as well as about the underlying communicative structure of the conversation. This kind of knowledge can serve the improvement of dialogue management systems.

The theoretical background of multimodal pragmatic annotation is largely based on Speech Act Theory (Bach & Harnish 1979) and represents its multimodal extension (Bódog, Abuczki, Németh T. 2011).

There are various types of communicative acts identified in the literature. Its typically distinguished classes (Bach & Harnish 1979) are the following:

- *constatives* express pondering, judging of the situation: giving a response, reinforcement, informing sy. about sg., prediction, remembering. Answers and confirmations are mostly uttered in declarative sentences.
- *directive* acts are those in which the speaker tells the addressee to fulfill some task (e.g. do a favour, follow an order, or accept a suggestion). Imperative or interrogative sentences can be devices of such acts.
- *commissives* are acts that aim at reaching an agreement (e. g. a bet), or the acceptation of a proposal. They force people to make a promise or accept something. Offers and promises are mostly uttered in the form of declarative sentences, too. The use of performative verbs (such as *promise*, *invite*, and *confirm*) indicate commissives.
- *acknowledgements* are greetings or acceptations (of invitations, for example). Utterances of greetings (e.g. *Jó napot kívánok!* = 'Good day') and accepting (e.g. *Köszönöm.* = 'Thank you') mostly occur in the form of declarative or imperative sentences, sometimes accompanied with a smile.

The label *none*, in our annotation, means non-identified/unidentifiable communicative acts. Besides, in acts labelled as *indirect* what is meant is something different from what is said in the utterance. The utterance expresses implicit, hidden, indirect meanings. For instance, the literal meaning and the intended meaning of the question *Can you pass the salt?* are different: literally, it is a yes/no question, but in fact, it is a request or a polite directive for action.

There are various acts supporting the ones listed above, as short iterative (repeated) phrases, like *jajaja; igen, igen, igen* or non-verbal signals like nodding that accompanies, among others, a constative act. These are called supportive acts. A type of supporting act is performed by the use of a *politeness marker* which is often a polite phrase (e.g. *légyszi* = 'please', with the communicative function of a smile). A reformulation or rephrasing is called *repair*, that is introduced by discourse markers such as *vagyis, vagy inkább* (='or rather'), *szóval, tehát* (='so'), or *pontosabban* (='more precisely', 'to be precise'). Such phrases can also be substituted or accompanied by hand gestures.

Turn-shifts (turn-taking, turn-giving and turn-keeping) are regulated in the framework of a so-called thematic control. By observing verbal content, we can judge how turns follow each other or to what an extent they fit into the framework of a conversation.

Starting a new topic, introduced by verbal markers such as *szerintem* (='I think'), is a *topic initiation*. Questions such as *Mit gondolsz erről/arról, hogy...?* ('What do you think about ...?') can have the same function. Sometimes these are accompanied by posture changes.

Topic elaboration is an event in which the initiated topic is detailed, evaluated or specified by verbal means. *Topic change* can be indicated by discourse markers, too, such as *egyébként* (='by the way'), *amúgy* (='otherwise') or *mellesleg* (='by the way'). A slight shift of the head, a look aside or a posture change can non-verbally accompany a topic change.

Signalling both giving and receiving new information is also an important condition of successful communication. The visual and verbal markers (with facial expression of *surprise* or verbal utterances such as *Aha*) representing this communicative function are to be identified in unimodal as well as multimodal annotation.

All in all, it was hoped to be illustrated in this chapter that the annotation of interpersonal interaction is an extremely complex and tedious task where all the aspects and modalities of communication have to be taken into account in order to grasp the intended and communicative meanings and functions expressed during the intricate course of the interaction.

References

Bach, K., Harnish, R. M. (1979): Linguistic commuication and speech acts. Cambridge: MIT Press.

Bódog A., Abuczki Á., Németh T. E. (2011): A multimodális pragmatikai annotáció jelentősége a számítógépes nyelvészetben. In: VIII. Magyar Számítógépes Nyelvészeti Konferencia. Szeged: Szegedi Tudományegyetem (240–251.)

Hunyadi, L., Földesi, A., Szekrényes, I., Staudt, A, Kiss, H., Abuczki, A. & Bódog A. (2012): Az ember-gép kommunikáció elméleti-technológiai modellje és nyelvtechnológiai vonatkozásai. In: Kenesei, I., Prószéky, G. & Várady T. (Eds.): *Általános Nyelvészeti Tanulmányok XXIV. Nyelvtechnológiai kutatások* (pp. 265–309). Budapest: Akadémiai Kiadó.

Kendon, A. (2004): Gesture. Visible Action as Utterance. Cambridge: CUP.

Massaro, D.W. (1987): Speech Perception by Ear and Eye: A Paradigm for Psychological Inquiry. Hillsdale, New Jersey: Lawrence Erlbaum.

Tools for Analysing Empirical Data: Doing Phonetics by Computer (Praat)

István Szekrényes

1) General presentation

The Praat[1] speech processing tool is an exemplary instance of how we can combine signal processing and linguistic studies. “Doing phonetics by computer” – as its own motto defines the goal, and the product has really become one of the most popular resource tools in this field. It is under continuous development and it applies widely accepted and well-documented speech processing methods. The program is quite easy to use even for those in the humanities who have limited experience in computer science. Although sound as a research material is just one of many kinds of data analysed in the humanities and, consequently, Praat is a rather specific tool, both are significant enough for us to focus on them in this chapter.

The Praat program can be used for various purposes. The commonly used functions are as follows:

- Acoustical analysis of speech
- Annotation (segmentation and labelling)
- Learning algorithms (neural networks, optimality theory)
- Graphics (e.g. figures for articles)
- Speech synthesis
- Listening experiments
- Speech manipulation
- Statistics

In this chapter, we cannot present every function in detail, but we try to provide a useful introduction for beginners about the most important features.

The software is a completely free, open source application and it is available for the following operating systems:

- Windows
- Macintosh
- Linux
- FreeBSD
- SGI

[1] Boersma, Paul & Weenink, David (2014). Praat: doing phonetics by computer [Computer program]. Version 5.3.62, retrieved 2 January 2014 from http://www.praat.org/

- Solaris
- HPUX

No installation is required (under Windows, Macintosh and Linux), you only have to download and unpack the ZIP archive from the official website[2] and run the executable file even from a flash drive. The source code is needed only if the user wants to modify something or add some extra features (for example, if somebody would like to use Python scripts in the programming interface[3]).

2) Functionalities of Praat

Work with the program can be performed in three different ways:

(a) by launching the graphical interface (Praat Objects and Praat Picture windows),
(b) by running Praat scripts from command line (Praatcon utility), or
(c) via external control (Sendpraat utility).

In the first case, you have to run the downloaded binary file, after what you are ready to use Praat Objects and Praat Picture windows. Under Macintosh and Linux systems, one can run Praat scripts from the command line (Windows users have to download the Praatcon[4] utility). Praat can also be controlled from an external application by sending messages with the **Sendpraat** utility. The syntax of the message is the following:

```
sendpraat [time of delay in seconds] praat|als [message]
```

The “message” part contains the command or commands for controlling Praat. There is a list of available commands with predefined “meanings”. For example, the following message containing the “Quit” command will stop Praat immediately:

```
sendpraat 0 praat Quit
```

The following (more advanced) example contains more than one command, which will be executed sequentially:

```
sendpraat 1000 praat "Read from file... MySounds/hello.wav" "Play reverse"
"Remove"
```

The first command („Read from file…”) has an argument, which defines the path of the sound file („hello.wav”) to be opened. 1000 seconds after sending the message, Praat will open the file, then play it reversely and then remove it from the list of objects. As one can see from the example, if a command contains an argument

[2] http://www.praat.org/
[3] https://github.com/JoshData/praat-py
[4] http://www.fon.hum.uva.nl/praat/download_win.html

one has to use quotation marks, otherwise the argument of the command is interpreted as a new command.

Some readers may get frightened of this seemingly complicated use of the command line interface, but as it was mentioned before, there is a **graphical interface** as well. The interface has two parts: the Praat Objects and the Praat Picture windows. The Praat Objects window enables the user to manage objects (e. g. sound files) in general, while one can visualize them or perform other graphical tasks in the Praat Picture window.

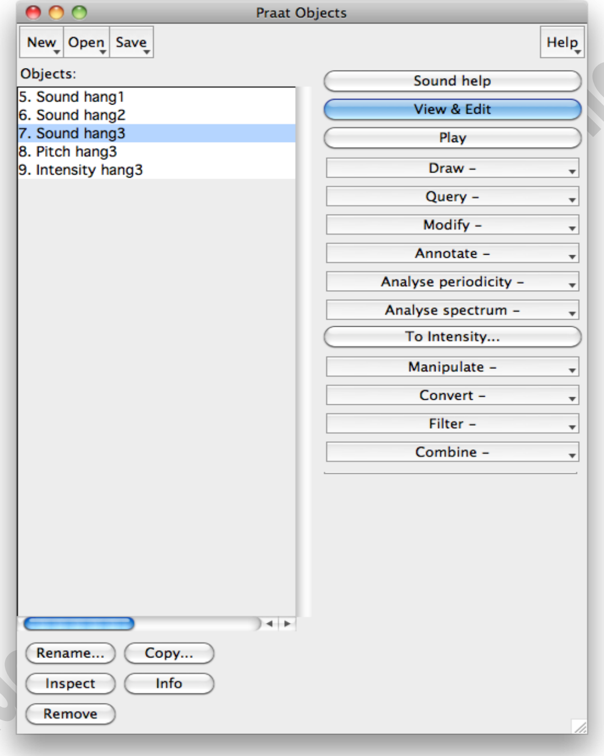


Figure 1: Praat Objects window (under Mac OS X)

Functionally, the **Praat Objects window** can be divided into four parts:

(a) Static menu (*Praat*, *New*, *Open*, *Save*)
(b) Dynamic menu (depending on the selected objects)
(c) List of objects (depending on the opened/created objects)
(d) Operational buttons (*Rename*, *Copy*, *Inspect*, *Info*, *Remove*)

The static menu (on the upper part, in horizontal layout) and the operational buttons (at the bottom) are continuously available and they provide invariable functions which are independent of the type of the selected objects. Obviously, the list of objects can be changed when you open, create or remove an object. The possible functions within the dynamic menu (on the right side, in vertical layout) depend on the type of the selected objects (in the case of Figure 1, these are the

Sound Help, *View & Edit*, *Play* etc. commands) just as every object type has its own storage format.

The objects in the list have three main properties:

(e) Identification number (ID): an object either created or opened has its own individual number
(f) Type: the list of available types can be checked out on the official website[5]
(g) Name: an arbitrary name (in case of opening, it will be the original name of the opened file)

In the list of Figure 1, these properties have the following values:

Table 1: Properties of the objects in Figure 1.

ID	Type	Name
5	Sound	hang1
6	Sound	hang2
7	Sound	hang3
8	Pitch	hang3
9	Intensity	hang3

We can perform different kinds of operations with operational buttons under the list. For instance, we can remove a selected object from the list with the "Remove" button; pressing the "Copy" button inserts a copy into the list (if we do not change the name, the ID will be the only difference between the copy and the original object).

The most important functions of the **static menu are**:

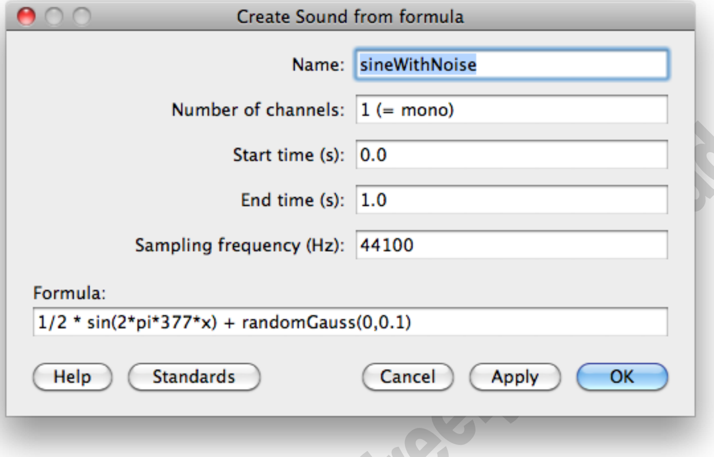
- Create an object (*New* menu)
- Open an object (*Open* menu – in earlier versions: *Read*)
- Write an object to the hard disk (*Save* menu - in earlier versions: *Write*)
- Open and run Praat scripts (*Praat* menu)
- Settings and others (*Praat* menu)

In a simple case, we can use the "Read from file…" command from the *Open* menu to open a file. The file will be placed into the list as a new object. Praat can automatically recognize the file format (if the format is supported) and assigns a type for the new object depending on the format (sound or annotation file, etc.). In Figure 1, the first three objects were identified as sound files, therefore they have the type property "Sound" as well. With the other commands in the *Open* menu we can also select specific types for the opened files. For instance, the "Open long sound file…" command reads sound files into the list with the "LongSound" type property instead of "Sound". The object with this "LongSound" type has limited

[5] http://www.fon.hum.uva.nl/praat/manual/Types_of_objects.html

processing possibilities, but it requires less physical memory that could eventually be useful for longer sound files.

New objects can be generated in the *New* menu. We can use type-dependent dialog boxes to set parameters for a new object. In Figure 2, we can see the dialog box of the “Create Sound from formula…” command.



Create Sound from formula

Name: sineWithNoise

Number of channels: 1 (= mono)

Start time (s): 0.0

End time (s): 1.0

Sampling frequency (Hz): 44100

Formula:
1/2 * sin(2*pi*377*x) + randomGauss(0,0.1)

Figure 2: Generating sound wave from formula

The “…”characters at the end of the command indicate that further arguments (“Name”, “Number of Channels”, “Start Time”, “End Time”, etc.) are needed. The values of these arguments can be set in the corresponding field of the dialog box. When using the command line or Praat scripts, we have to list the parameters in the same sequential order separated by spaces:

```
Create sound from formula… “sineWithNoise” 1 0 44100 “1/2*sin(2*pi*377*x) +
randomGauss(0,01)”
```

If we use certain kinds of commands from the **dynamic menu**, new objects can be generated from the selected original object and inserted into the list. In case of “Sound” type objects, the “To Pitch…” command (in the “Analyse periodicity” group) creates a new object with a new, “Pitch” type and with the same name as the original “Sound” object. This “Pitch” object will also have the same timing and it will contain the measured pitch (F0) data from the “Sound” object (in Figure 1, the “Pitch hang3” was created in this way from “Sound hang3”). Owing to these generating functions, we can investigate suprasegmental features of speech within a separate object.

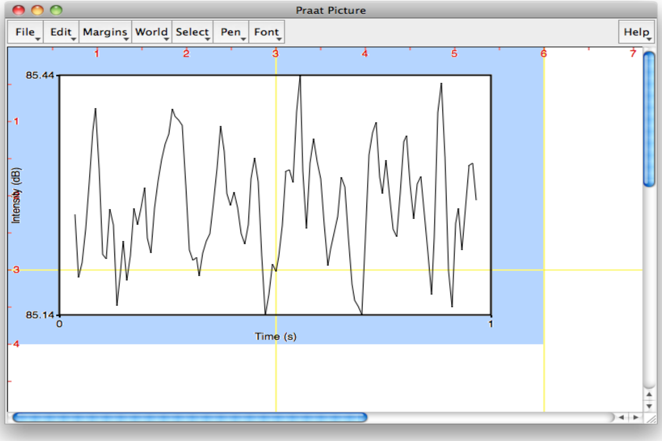
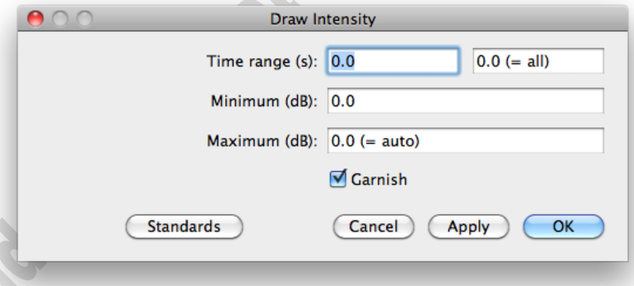


Figure 3: Praat Picture window

The **Praat Picture window** can be controlled within its own menu system and from the Praat Object window using the "Draw" commands, available for some of the object types in the dynamic menu. For instance, the curve in Figure 3 is the visualisation of the "Intensity hang3" object from the list of Figure 1. The picture is generated by the "Draw" command, which can be set-up in the following dialog box (see Figure 4).


Draw Intensity

Time range (s): 0.0 | 0.0 (= all)

Minimum (dB): 0.0

Maximum (dB): 0.0 (= auto)

☑ Garnish

Figure 4: Dialog box of drawing intensity

Further possible commands are available in the Praat Picture window's own menu system. They are suitable to control the drawing interface:

- File menu: open/save figures and graphs
- Edit menu: withdraw, copy, erase
- Margins menu: set the margins of graphs
- World menu: drawing (define and draw lines or functions)
- Select menu: set the size and other properties of the drawing surface
- Pen menu: set drawing lines (color, width etc.)
- Font menu: set font properties

Praat also has various kinds of **editor windows** which can be displayed from the dynamic menu with the "View & Edit" command if available for a given object type. The functions and the menu system of the editor windows can vary depending on the type of objects. In Figure 5, a “Sound” type object’s editor window can be seen:

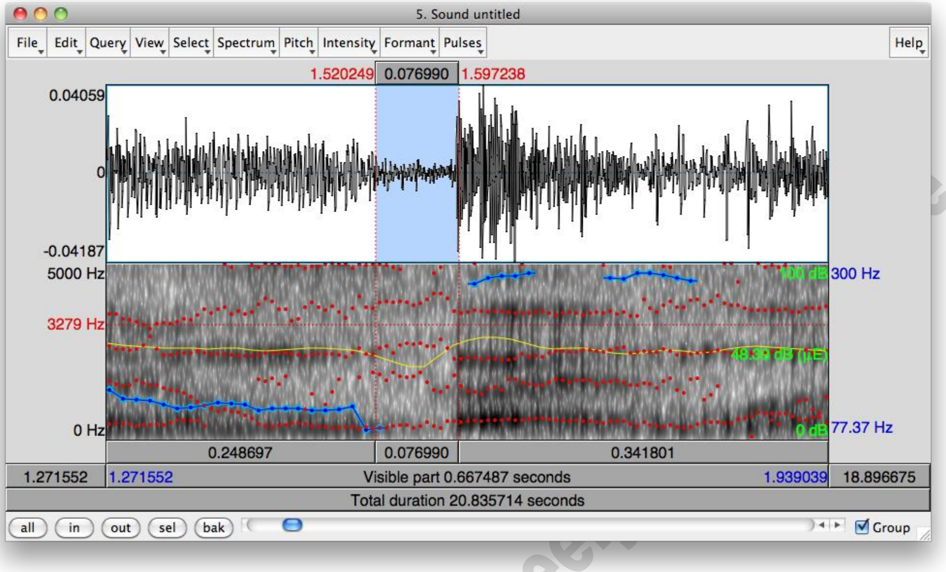


Figure 5: Editor window of “Sound” objects

There are navigation buttons and a slider at the bottom (for scrolling and zooming). The three grey bars above them are for playing the sound: clicking on the lowermost bar plays the entire file, the middle one plays the content of the entire window, the top one plays the selection. The same functions are also available from the “View” menu, and, alternatively, we can also use hotkeys (e. g. “Ctrl + N” for zooming to the selection).

From the file menu, we can perform various kinds of operations on the selected part of the sound:

- Create a new object from the selection: “Extract selected sound…”
- Save the selection to the hard disk: “Save selected sound…”
- Draw the selection into the Praat Picture window: “Draw selected sound…”

We can set the selection and the position of the cursor in the “Select” menu (of course, we can use the mouse as well), and we can query this information in the “Query” menu (e.g. “Get selection length”). The rest of the menus in Figure 5 contain object-specific (type dependent) commands. In case of the “Sound” type, we can visualise the following prosodic features (turning on one of the “Show…” commands):

- Spectrogram (Spectrum menu): represented by the audio spectrum under the wave form

- Fundamental frequency (Pitch menu): represented by the blue curve
- Intensity (Intensity menu): represented by the yellow curve
- Formants (Formant menu): represented by the red points

In these menus, we can perform further operations with these features:

- Set-up the visualization properties
- Query data related to the feature (values at certain time)
- Draw the selected part of the signal into the Praat Picture window

The function of the Praat Info window is to display the results of queries from Praat Objects and the editor windows. Praat scripts also use this window as a standard output. It has its own menu system where we can save the content of the window and use other commands (Undo, Redo, Copy, Paste, Find, Replace etc.).

The most important function of Praat is its well-known **acoustical analysis of speech**. This function is supported by largest number of object types. After opening, generating or recording a sound file, a wide range of commands (e.g. "To Pitch...", "To Intensity...", "To PitchTier..." etc.) are available from the dynamic menu of the Praat Object window to investigate the prosodic features of speech. Some of the available object types and the possible routes to generate them from other types are shown in Figure 6.

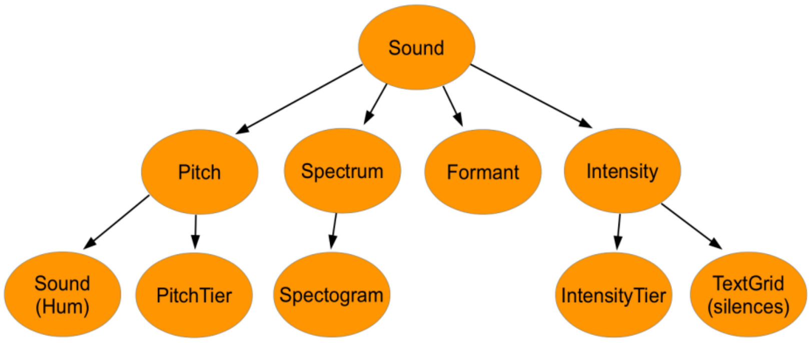


Figure 6: Generation of speech acoustic objects in Praat

We can also investigate these generated objects in an editor window by using the “View & Edit” command.

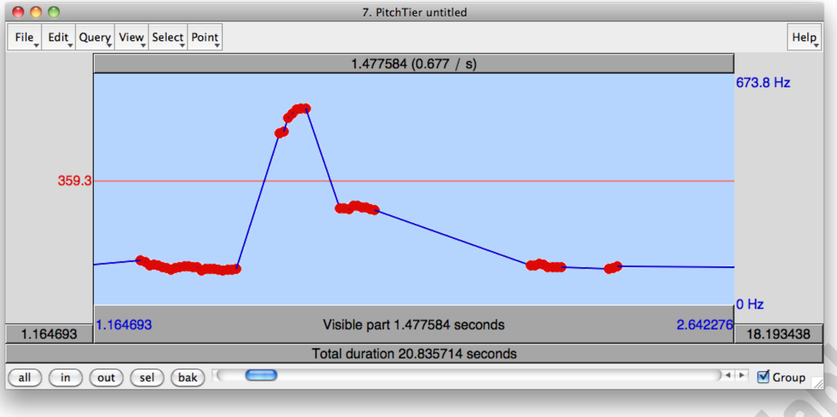


Figure 7: Editor window of a "PitchTier" object

The editor window in Figure 7 belongs to a „PitchTier" object which contains F0 data with time and frequency values. One can query the data even from the dynamic menu, and these query commands can be included in a Praat script to generate data tables for statistical analysis. The objects can be saved as plain text files in various formats. For instance, one can store a "PitchTier" object in the following format:

```
File type = "ooTextFile"
Object class = "PitchTier"
xmin = 0
xmax = 20.835714285714285
points: size = 605
points [1]:
number = 0.29285714285714237
value = 79.71932258134986
points [2]:
number = 0.3028571428571424
value = 78.9987075127485
points [3]:
```

Praat has further complex, but very useful built-in procedures for speech analysis. For instances, we can automatically detect silences and generate an annotation ("TextGrid") object from the sound file containing the boundaries of silences as well as sounding parts of the speech ("To TextGrid (silences)…" command).

Annotation is also a very useful and important function of the program. One can use the "TextGrid" type to create an annotation object, selecting the "To TextGrid…" (dynamic menu) or the "Create TextGrid…" (*New* menu) commands. We can use a specific editor window for annotation by simultaneously selecting the "TextGrid" and the "Sound" files and then clicking the "View & Edit" button. During the annotation, various kinds of labels can be created for the annotation of the speech signal including its segmentation and the description of certain speech properties. The process of annotation may be manual, based on human observation as well as automatic, produced by an algorithm (e.g. by running a Praat script). Creating an orthographic transcription (a kind of a “subtitle”) for speech is the most typical case of annotation, but it may consist in marking disfluencies or spans of silence (as mentioned above) as well.

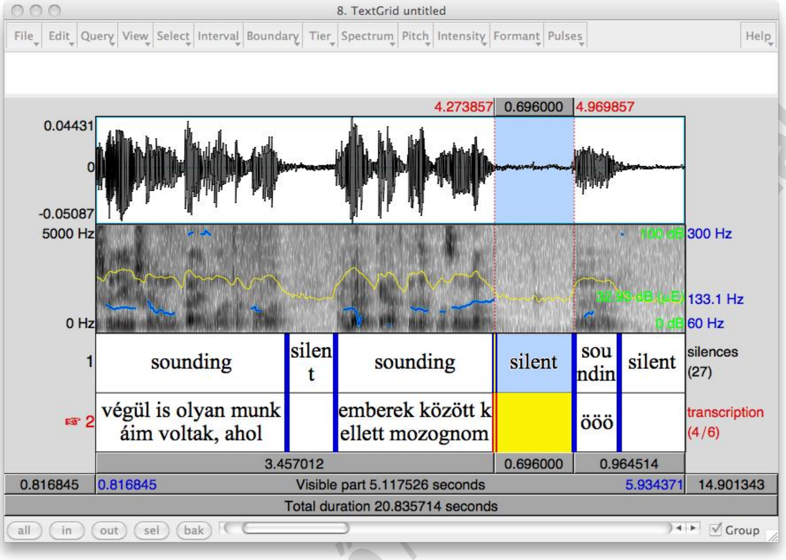


Figure 8: Annotation in Praat

Figure 8 above shows two annotation tiers (“silences” and “transcription”), but further tiers can be added if we want to broaden our investigation. The blue vertical lines represent the boundaries of annotation segments. Boundaries can be added or removed in the “Boundary” menu, and the annotations can be also managed using the commands of the dynamic menu. For instance, queries can be performed on the annotation segments (“Get start time…”, “Get label of interval…” etc.).

Praat has a built-in **scripting language** (Praat script) which is a very powerful tool for advanced users to go beyond the predefined functions of the menus. The Praat script is a complete, imperative, procedure-oriented programming language that uses the Praat program as an interpreter and execution environment. The most important feature of the language is that it offers the possibility to use all the menu commands (Praat functions) in command line form like with the Sendpraat utility and thus create a complex code of a structured sequence of functions . With the code created this way, one can run the script in a special window opened with the commands “New Praat script” or “Open Praat script” from the “Praat” menu. There is an excellent English documentation of Praat scripting on the official website[6], therefore we only provide a short example in this chapter.

```
select all
n = numberOfSelected ("Sound")
for i from 1 to n
```

[6] http://www.fon.hum.uva.nl/praat/manual/Scripting.html

```
    sound'i' = selected ("Sound",i)
endfor
for i from 1 to n
    select sound'i'
    name$ = selected$ ("Sound")
    do ("To Pitch...", 0, 75, 600)
    f0mean = do ("Get mean...", 0, 0, "Hertz")
    if f0mean > 150
        appendInfoLine ("The speaker of 'name$' sound is probably female")
    else
        appendInfoLine ("The speaker of 'name$' sound is probably male ")
    endif
endfor
```

The above script is a quite simple and generic solution for detecting the gender of the speaker in sound files containing only one speaker. By running the script, the same sequence of operations is executed on every selected “Sound” object. The script first extracts the F0 values from the given sound file by generating a “Pitch” object, and then it calculates the average of these values in every object. If the average is above 150 Hz, a message appears in the Praat Info window: “The speaker of [name of the current object] sound is probably female”. If the average is under 150 Hz, the message is different: “The speaker of [name of the current object] sound is probably male”.

In summary: the Praat program is a highly usable tool for the comprehensive analysis of empirical data based on speech. By analysing frequency, intensity, duration and other complex categories further derived from the latter, one can reach deeper in the study and understanding of human communication in general. It is a tool advancing our empirical knowledge based on language and beyond; a freely accessible research environment everyone with an interest in speech processing is invited to get acquainted with.

An Overview of Multimodal Corpora, Annotation Tools and Schemes

Ghazaleh Esfandiari, Ágnes Abuczki

The aim of the present study[1] is to provide a general overview of multimodal corpora along with its annotation tools and schemes. In the first section an introduction is given to justify the necessity of multimodal approach not only in linguistics and communication studies but also in dialog modeling. Following the justification and the definition of multimodal corpora, three relatively well-known corpora are presented and compared. The considerable differences in their platforms and schemes highlight the necessity of standardization in order to enhance the usability and interoperability of their resulting datasets. Finally, the limitations of multimodal corpus studies are briefly listed.

1) Introduction

Corpus linguistics (henceforth: CL) studies natural language use, that is, performance, in contrast with theoretical linguistics where the focus is on the study of competence. It can be defined as a methodological research approach to studying language variation and use (Biber & Reppen 2012). Since CL uses a large and principled collection of natural texts as a basis for quantitative and qualitative as well as traditional and computational analyses, its research findings have great generalizability and validity. The goals of CL include providing explicit descriptions of language-in-use, finding sequential patterns in language, verifying models of language use and developing algorithms in various fields of language technology.

Methodological innovations in CL allow linguists to raise fundamentally different kinds of research questions, sometimes providing different perspectives on language variation and language use (Biber & Reppen 2012). For instance, as computing capacities began to increase in the 1970s, CL research interest shifted from the study of written to spoken interaction. It was uncovered with the help of CL that spoken language greatly differs from written text since not all sentences are well-formed, sentence boundaries are often unclear, and that different principles and maxims guide spoken interaction. All these findings led to the necessity of novel units of segmentation in spoken corpora, such as turns or utterances. However, as Knight et al. (2006) argue, spoken corpora are limited as they only

[1] First appeared in *Argumentum* as Abuczki, Á. & Esfandiari B. G. (2013) "An overview of multimodal corpora, annotation tools and schemes". *Argumentum*, 9, 86–98.

have the provision for presenting data in a single format; that is text, in the form of transcripts of interactions. Spoken corpora with transcripts alone are not sufficient for uncovering the nonverbal-visual aspects of interaction since ‘the reflexivity of gesture, movement and setting is difficult to express in a transcript’ (Saferstein, 2004: 213). As a result, CL research interest has recently shifted to the study of multimodal interaction and its capturing in corpora.

Multimodal corpus (henceforth: MM corpus) research faces two major problems: (1) the time consuming nature of annotation process, therefore, relatively small sizes of annotated MM corpora are available; (2) the lack of existing annotation standards (tools, formats and schemes), especially for coding nonverbal behavior. There are several annotation schemes available that code the nonverbal behavior of speakers; however, most of them do not integrate talk and gesticulation in a coherent fashion. It would be crucial at least to recognize the transitions between intentional and non-intentional movements (e.g. the difference between a real gesture and an accidental movement).

2) The necessity of a multimodal approach in communication studies

The aim of this section is to highlight the multimodal nature of human expression and perception that must be considered in designing corpora. We attempt to briefly answer why we need a multimodal approach in studies of language use. In general, most multimodal studies address the questions of how modalities work together as well as to what an extent verbal and nonverbal expressions are synchronized.

a) The multimodal nature of human interaction

McNeill (1992: 30–32) proposes that gestures must be regarded as part of language: speech and gesture are two modes of expression and they can be regarded as two aspects of the process of an utterance. Similarly, Kendon (2004) argues that a theory of utterance should not begin with a division between speech and gesture. In Kendon’s definition, a gesture is a form of human expression, an activity that is significant for the understanding of a speaker’s expression, and they often express something complementary to what is being expressed verbally (2004: 107).

b) Multimodal perception

Massaro (1987) points out that it is not only our expressions but also our perception that is of multimodal nature since we process not only what we hear but also what we see (facial expressions, lip movements, eyebrow movements, hand gestures of speakers, etc). Inherently, there is both auditory and visual relation among speakers in which the modalities normally complement, sometimes contrast each other. Simultaneous auditory and visual perception is called multimodal perception, and the processing of stimuli coming from various modalities is largely an unconscious process (Massaro 1987). Since auditory and visual perception is both simultaneous and unconscious, the nonverbal components of interaction must also be taken into account when interpreting a communicative event.

3) The definition and requirements of MM corpora

A ‘multi-modal corpus’ is defined as ‘an annotated collection of coordinated content on communication channels including speech, gaze, hand gesture and body language, and is generally based on recorded human behavior’ (Foster & Oberlander, 2007: 307–308). The integration of textual, audio and video records of communicative events in MM corpora provides a platform for the exploration of a range of lexical, prosodic and gestural features of conversation, and for investigations of the ways in which these features interact in real, everyday speech (Knight 2011: 15).

Within various types of MM corpora, we can distinguish two basic types:

- video recordings supplemented with only transcriptions;
- video and audio recordings annotated at multiple levels (based on both audio and video separately).

All three corpora presented in this study belong to the second category which is considered more valuable in communication studies.

Biber & Reppen (2012) list the following requirements of corpora:

- representativity
- validity
- generalizability
- standardized

We would also complement this list in connection with MM corpora that their annotation schemes should be domain and tool-independent, and their labels (within a single level at least) should be mutually exclusive. Moreover, beside its audio and video contents, a usable MM corpus must also have metadata description, annotation guidelines and users guide in order to provide rigorous guidelines to its coders as well as to ensure its usability for researchers.

2) Annotation tools and query options related to MM corpora

a) Annotation and querying tools

Generally, different annotation tools are designed and used to annotate the audio and video contents of a corpus that can later be merged in query systems or databases. For instance, video contents of the HuComTech corpus were annotated in Qannot (Pápay et al. 2011: 330–347), an environment specifically designed for our purposes, while audio contents were annotated in Praat, a fine grained audio analysis tool (Boersma & Weenink 2007) which enables a much more precise and detailed acoustic analysis than compact multimodal annotation software such as Anvil (Multimodal Annotation and Visualization Tool[2]) or ELAN (Brugman & Russel 2004: 2065–2068). However, Anvil and ELAN offer a lot of benefits to its users since they enable the simultaneous streaming and annotation of both audio and (even multiple) video files in separate windows, and users can specifically design their own annotation scheme and attach multiple tags to one segment in both pieces of software. Moreover, Anvil allows multiple annotators to work on the same file and therefore it is able to measure inter-annotator agreement. Concerning the video annotation tool of the HuComTech corpus, a new software, Qannot was designed instead of Anvil because Anvil sometimes seemed to fail to handle large files and there was a risk that the timestamps of annotations might be inaccurate in these large files. As the annotations were complete, the various annotation files of the HuComTech corpus were merged in an SQL database. Annotations are still stored in SQL and can also be queried in a very user-friendly way using the ELAN software (Brugman & Russel 2004). Custom query options of ELAN include: *N-gram within annotations; Structured search of multiple files; Find overlapping labels within a file;* and *Find left overlaps within a file*, etc. The availability of multimodal annotation tiers enables the systematic and joint search of the temporal alignment and/or synchronous co-occurrences of turns, clauses or specific lexical items with the use of manual gestures, head movement types, gaze directions, eyebrow movement types and posture changes in spontaneous interaction corpora.

b) Usability of datasets in novel corpus-driven research areas

With the help of MM corpora searches, the investigation of the temporal alignment (synchronized co-occurrence, overlap or consecutivity) of gesture and talk has become possible. Similarly to corpus-driven approaches that study *lexical bundles* (multi-word sequences) (Biber 2010: 170–172), some of the MM corpus researches are inspired by the notion of *semiotic bundles* (Arzarello et al. 2005) where modelling language production includes the manipulation of resources as well as gesture and talk. Some functional annotation schemes (Allwood et al. 2007) try to code the meaning relations between gestures and co-occurring speech in a systematic way, and label communicative events according to the alignment of speech and gesture. Gestures often co-occur with speech; however, their discursive functions are not always identical. The basic function of the gestures and speech either ‘overlap’ or are ‘disjunct’, and sometimes synchronous verbalisations and gestures may be more ‘specific’ than the other sign at a given timestamp in the annotation (Evans et al., 2001: 316). Frequency evidence (of any sequential linguistic pattern and co-occurring nonverbal phenomena) found in corpora supports the application of statistical methods in language analysis and modelling.

[2] ANVIL is freely available at: http://www.anvil-software.org/

The huge amounts of synchronized data enable the practical and fruitful use of such advanced statistical methods as factor analysis or multidimensional analysis in order to uncover the prototypical features that simultaneously occur in certain communicative acts. Therefore, these methods contribute to the solution of a challenging task in dialog modelling and dialog management, the automatic identification of dialog structure and communicative act types.

3) Examples of MM corpora

This section aims at providing a general overview of MM corpora by describing a few examples. The corpora chosen for this purpose are AMI, SmartKom and HuComTech. These three different corpora were chosen in a way to represent the variety of approaches and aims involved in structuring MM corpora. Therefore, they can be contrasted in terms of their different types of discourse following different scenarios, such as meetings, task-based interaction, simulated job interviews and informal conversations. In the following section, each one of them will be described briefly, providing their particular aim, context of use, structure and annotation scheme.

a) AMI Corpus

i) Aim and Context of Use

The AMI or Augmented Multi-party Interaction Corpus is a large MM corpus, involving 100 hours of meetings. Its aim was to develop and integrate meeting browsing technologies in order to support human interaction in meetings. The corpus focuses on language use in a single setting, which is a meeting room, so it is contextually specific and it only features extracts from one specific discourse context (i.e. meeting discourse) thus its usefulness is limited in studying more informal, interpersonal aspects of language use (Carletta et al. 2005).

ii) Corpus Design

While some of the meetings in this 100-hour long corpus were naturally occurring (35 hours), the majority (65 hours) was elicited using a scenario in which groups of three to four participants played different roles as employees working on a design project in a design team. The data was collected in three smart meeting rooms. In each room 4 cameras, 24 microphones and special tools to capture handwriting and slides were used (McCowan et al. 2005). The language of communication in all meetings was English while most of the participants were non-native English speakers. Due to this fact, a higher degree of variability in speech patterns can be observed in this corpus compared to other corpora.

iii) *Annotation Scheme*

The data has been annotated at a number of levels covering various verbal and nonverbal features. Table 1 summarizes the annotation scheme used in this corpus (Carletta et al. 2005).

Levels of annotation	Annotated elements
Speech transcription	orthographic transcription of speech, also annotating speaker change boundaries and word timings
Named entities	reference to people, artefacts, times and numbers
Dialogue acts	act typology used for group decision-making
Topic segmentation	major topic and sub-topic segments in meetings
Group activity	activities that groups are engaged in
Abstractive summaries	decisions that were made during the meeting, problems or difficulties that occurred during the meeting, next steps
Extractive summaries	extract a subset of the dialogue acts of the meeting, such that they form a kind of summary and then link those extracted dialogue acts with sentences
Emotion	different dimensions which reflect the range of emotions that occur in the meetings
Head and hand gestures	movements of both the head and the hands of the participants
Location of the individual	location of the individual in the room or the posture if seated
Focus of attention	what the participants are looking at (which people or artifacts)

Table 1. Annotation scheme used in AMI corpus

The AMI Meeting Corpus is publicly available at http://corpus.amiproject.org containing media files (audio files, video files, captured slides, whiteboard and paper notes) and also all annotation dimensions described in Table 1. However, the annotated dimensions as well as the implicit metadata for the corpus are difficult to exploit by NLP tools due to their particular coding schemes.

b) SmartKom Corpus

i) *Aim and Context of Use*

The SmartKom corpus was built as part of the SmartKom project in Germany with the goal to develop an intelligent computer-user interface allowing for more natural interaction for users. SmartKom is one of the first corpora that combines the analysis of acoustic, visual and tactile modalities. It is a task-oriented corpus since that data were gathered and annotated having specific aims and has therefore a limited re-usability for other purposes (Schiel et al. 2002).

ii) Corpus Design

The data was gathered using so called Wizard-of-Oz experiments. In this experiment, participants were asked to work on a specific task while cooperating with the system. The subjects thought that they were really interacting with an existing system but in reality the system was simulated by two humans from another room. 96 different users were recorded across 172 sessions of 4.5 minutes each. In each Wizard-of-Oz session, spontaneous speech, facial expressions and gestures of the subjects were recorded and later annotated. The language of communication was German in all recorded sessions (Steininger et al. 2002).

iii) Annotation Scheme

The data has been annotated on several levels covering various features. Table 2 summarizes the annotation scheme used in this corpus (Steininger et al. 2002). This corpus is available for academic use only through the META-SHARE website[3]. (META-SHARE is an international organization which builds a multi-layer infrastructure and aims at providing an open, distributed, secure, and interoperable infrastructure for the language technology domain.) Release SKAUDIO 1.0 contains all audio channel recordings of the SmartKom corpus covering all three scenarios (Public, Home and Mobil) used in the technical setup.

Levels of annotation	Annotated elements
Speech transliteration	orthographic transliteration on word level of spontaneous dialogue between user and machine
Head gestures	three morphological categories, head rotation, head incline forward/backward, head incline sideward
Hand gestures	functional and intentional (not morphological), based on the intention of the user's assumed goal
Emotional facial expressions	joy/gratification, anger/irritation, helplessness, pondering/reflecting, surprise, neutral, unidentifiable episode
Prosody	pauses between phrases, words and syllables, irregular length of syllables, emphasized words, strongly emphasized words, clearly articulated words, hyper articulated words, words overlapped by laughing

Table 2. Annotation scheme used in SmartKom corpus

The annotation of the nonverbal-visual components of interaction in both AMI and SmartKom is somewhat incomplete and inapplicable for an in-depth analysis

[3] META-SHARE website: http://www.meta-net.eu/meta-share

of interpersonal communication since they both predominantly aim at capturing movements and fail to label the visual features with their meanings or functions in the particular discourse context. For instance, AMI annotates movements of the head and the hands of the participants and SmartKom annotates head gestures based on three morphological categories, head rotation, head incline forward/backward, head incline sideward. At the same time, we can find alternative annotation schemes among MM corpora which try to integrate talk and gesticulation in a coherent, truly multimodal scheme, such as MUMIN (A Nordic Network for MUltiModal INterfaces) developed by Alwood et al (2007) or HuComTech (described in Section 5.3 below and in Hunyadi et al. 2012a in detail).

c) HuComTech Corpus

i) *Aim and Context of Use*

The MM HuComTech corpus was built in the framework of the Human-Computer Interaction Technologies project[4]. Hungarian was the language used in all recorded conversations. The aim of building the corpus was to investigate the nature and temporal alignment of verbal and nonverbal features of spontaneous speech as well as to compare the characteristics of formal and informal communication as the corpus involves both formal and informal conversations (between dialogue partners). It is useful to include two types of conversation, formal and informal for purposes of comparative analysis since formal conversations follow rules and strong social norms and involve the use of keywords, symbolic gestures, high conscious control, while the structure and scenario of informal conversations are not so strict (overlapping turns, inconsistencies, discrepancies between modalities, iconic gestures, other eventualities often occur). This distinction is important for the sake of defining spontaneity within interaction, and drawing our technological limits (Pápay et al. 2011).

ii) *Corpus Design*

The material contains 50 hours of both formal and informal dialogues from 121 speakers. The dialogues were recorded in a soundproof studio. The participants were both audio and video taped during their conversations. The informal dialogues centred on everyday topics, mostly about university and other life experiences while formal dialogues followed the typical scenario of simulated job interviews. Both the formal and informal dialogues were guided by pre-designed questions that intended to provoke various emotions such as happy, sad, angry and surprised (Pápay et al. 2011).

iii) *Annotation Scheme*

The data was annotated on different levels coding various features. The annotation was carried out based on either one modality (audio only or video only) or two modalities (audio and video). This corpus also includes syntactic, prosodic

[4] HuComTech website: https://hucomtech.unideb.hu/hucomtech/

and pragmatic annotation. The syntactic annotation was restricted to the identification and classification of clauses and sentences (Hunyadi et al. 2012a). In the prosodic annotation, the F0 and intensity movements were annotated (Hunyadi et al. 2012b). Table 3 and 4 briefly summarize the annotation schemes used in this corpus.

Levels of annotation	**Annotated elements**
Speech transcription	orthographic transcription of speech for both speakers
Discourse labels	turn take, turn give, turn keep and backchannels
Emotions	happy, tense, sad, recall, surprise, neutral, other
Intonational phrases	head clause, subordinate clause, embedding, insertion, back channel, hesitation, restarts, iterations and silence

Table3. HuComTech annotation scheme based on audio-only

Levels of annotation	**Annotated elements**
Facial expressions	happy, tense, sad, recall, surprise, neutral, other
Gaze	gaze direction of the speaker using various directional labels
Eyebrow	movement of the speaker’s eyebrow using various directional labels
Head shifts	movement of the speaker’s head using various directional labels
Hand shape	shape of the speaker’s hand
Touch motion	the speaker touching one or some of his/her body parts
Posture	body shifts of the speaker using various directional labels
Deictic	the speaker points at him/herself or something else present in the room
Emotion	happy, tense, sad, recall, surprise, neutral, other
Emblems	attention, agree, doubt, disagree, refusal, block, doubt-shrug, finger-ring, hands-up, more-or-less, number, one-hand-other-hand, surprise-hands and other

Table4. HuComTech annotation scheme based on video and audio

The pragmatic annotation was carried out on two separate levels, multimodal (based on both audio and video) and unimodal (based on video only), the latter being a novel approach in pragmatic corpus annotation.

Multimodal pragmatic annotation codes communicative functions and speaker intentions, not necessarily mirrored in surface structure. For instance, an interrogative sentence may express a directive function. The major aim of the multimodal pragmatic annotation was to find the underlying structure of communicative behavior as well as the visual, acoustic and verbal correlates of different communicative acts (Abuczki et al. 2011: 179–201).

As for the unimodal annotation, the aim was to grasp communicative events based solely on visual input. Table 5 and 6 outlines the pragmatic annotation schemes used in this corpus.

Levels of annotation	**Annotated elements**
Communicative act types	constative, directive, commissive, acknowledging and indirect
Supporting acts	backchannel, politeness marker and repair
Thematic control	topic initiation, topic elaboration and topic change
Information	units of new information

Table 5. HuComTech multimodal pragmatic annotation scheme

Levels of annotation	**Annotated elements**
Turn management	start speaking successfully, breaking in, intend to start speaking and end speaking
Attention	call attention, pay attention
Agreement	agreement and disagreement and its degree: default case of agreement, full agreement, partial agreement, uncertainty, default case of disagreement, blocking and uninterested
Deixis	deictic gestures not annotated in the video annotation
Information structure	received novelty was annotated

Table 6. HuComTech unimodal pragmatic annotation scheme

This corpus is not publicly available yet. It is available for academic use only through the META-SHARE website.

4) Standardization

In the previous section a brief overview of three different MM corpora were provided. These corpora differ with respect to their approaches and annotation schemes as well. In each one of them, different nonverbal behaviours were selected and annotated using different labels defined in specific ways serving their own purpose of study. Therefore, the design of a MM corpus could not rely on conventionalized prescriptions that determine which behaviours to mark-up, how to describe these behaviours, which labels to use in the annotation scheme and how to integrate everything in the corpus database to cover all multimodal elements of discourse. As a result, generalizing standards for codification of visual and spoken data should be considered as a priority in multimodal research (Knight 2009). Recently, many researchers and research teams have started to lay the foundations for designing a standardized scheme for annotating various features of spoken utterances, gaze movement, facial expressions, gestures, body posture and combination of any of these features. They have the aim to integrate these aspects to develop re-usable and international standards for investigating language and gesture-in-use in user-friendly environments. The outcome of such international interdisciplinary initiations and cooperation are for instance the META-SHARE, the HUMAINE[5] (Human-Machine Interaction Network on Emotion) and the SEMAINE[6] (The Sensitive Agent) projects. The HUMAINE project developed the XML-coded EARL (Emotion Annotation and Representation Language) scheme[7] to annotate the dimensions and intensity of emotions. However, it can only be used with the Anvil software. Its restricted usability highlights the necessity of tool- and domain-independent annotation schemes.

The SAIBA project developed the tool- and domain-independent Behaviour Markup Language (BML) (Vilhjalmsson et al. 2007). BML is a widely used method to unify the key interfaces in multimodal human behaviour generation processes. ISO standard 24617-2 for dialogue acts developed in recent years is an example of a widely accepted international standard (Bunt et al. 2012). It is an application-independent dialogue act annotation scheme that is both empirically and theoretically well founded. It covers typed, spoken, and multimodal dialogue, and it can be effectively used by both human annotators and automatic annotation methods. In designing this ISO standard for dialogue act annotation, most concepts were applied from the DIT++ taxonomy of dialogue acts[8]. Table 7 summarizes the annotation scheme used in this ISO standard.

[5] HUMAINE: http://emotion-research.net/projects/humaine/aboutHUMAINE

[6] SEMAINE: http://www.semaine-project.eu/

[7] EARL-scheme: http://emotion-research.net/projects/humaine/earl

[8] DIT++ taxonomy is available at http://dit.uvt.nl/

General-purpose functions	**Information-seeking functions**: *propositional questions, check questions, set questions and choice questions*
	Information-providing functions: *inform, agreement, disagreement, answer, confirm and disconfirm*
	Commissive functions: *promise and offer*
	Directive functions: *instruct/command and request*
Dimension-specific functions	**Auto-feedback functions**: *positive and negative*
	Allo-feedback functions: *positive and negative*
	Time management functions: *stalling and pausing*
	Turn management functions: *turn accept, turn assign, turn grab, turn keep, turn release and turn take*
	Discourse structuring functions: *interaction structuring and opening*
	Own and partner communication management functions: *completion, correct misspeaking, signal speaking error, retraction and self correction*
	Social obligation management functions: *initial greeting, return greeting, initial self introduction, return self introduction, apology, accept apology, thanking, accept thanking, initial goodbye and return goodbye*

Table 7. ISO standard for dialogue act annotation scheme

5) Limitations

This section aims at outlining some of the major limitations in multimodal corpus research. First, each corpus with its design and annotation scheme serves specific purposes. Therefore, they do not cover all elements and types of communicative events. Most gesture annotation schemes only describe the size, trajectory and direction of movements that gestures are comprised of. Some of them are restricted to the analysis of only one large component of human nonverbal expressions according to kinesic properties, such as the detailed analysis of face by the FACS scheme (Facial Action Coding System) (Ekman et al. 2002). Another movement-based annotation scheme comes from McNeill (1992) who labels a variety of manual gestures as well besides the facial expressions and head movements, but it still fails to integrate talk and gesticulation in a truly multimodal framework. Second, the relatively small size of all corpora (tens or hundreds hours of annotated interaction) makes the observation of all sorts of linguistic phenomena in a natural proportion impossible (Knight 2009: 99–100). Third, participants are typically seated that highly restricts their movements and posture changes. Fourth, the number and positioning of cameras restrict the size of analysable environment related to each speaker. Frequently, each of the speakers is recorded by different cameras, which makes it impossible to investigate such phenomena as joint attention or eye contact. Synchronization of the cameras and their joint streaming in annotation software such as ELAN can be an acceptable solution in these cases. Last but not least, we must mention that annotation (for instance, pragmatic annotation) involves interpretation to some extent, so it is never completely free of intuitions. However, this problem can be handled if the annotation guidelines are consistently followed and if annotators regularly attend meetings where ambiguous cases can be discussed.

Although we are aware of the existing shortcomings of MM corpora (Sinclair, 2008: 30), we believe there is no better alternative resource available for analysing spontaneous language-in-use than a MM corpus, since current corpus linguistics provides us various with methodologies to process and analyse interaction in its entire multimodal nature.

References

Abuczki, Á., Bódog, A. & Németh T., E. (2011): A pragmatikai annotáció elméleti alapjai az ember– gép kommunikáció modellálásában. In: Németh T., E. (ed.): Ember–gép kapcsolat: A multimodális ember–gép kommunikáció modellezésének alapjai. Budapest: Tinta Kiadó, 179–201.

Abuczki, Á. & Esfandiari B. G. (2013). An overview of multimodal corpora, annotation tools and schemes. *Argumentum*, 9, 86–98.

Allwood, J., Cerrato, L., Jokinen, K., Navarretta, C., & Paggio, P. (2007): The MUMIN coding scheme for the annotation of feedback, turn management and sequencing phenomena. Language Resources and Evaluation 41(3-4), 273–287.

Arzarello, F., Ferrara, F., Paola, D. & Robutti, O. (2005). The genesis of signs by gestures. The case of Gustavo. In H. L. Chick & J. L. Vincent (eds.), Proc. of the 29th Conference of the International Group for the PME, 1. Melbourne, AU: University of Melbourne, 73–83.

Biber, D. (2010): Corpus-based and corpus-driven analyses of language variation and use. In: Heine, B. & Narrog, H. (eds.): The Oxford Handbook of Linguistic Analysis.

Biber, D. & Reppen, R. (2012): Corpus Linguistics. SAGE Publications Ltd.

Boersma, P. & Weenink, D. (2007): Praat: doing phonetics by computer 5.0.02. University of Amsterdam: Institute of Phonetic Sciences. http://www.praat.org

Brugman, H. & Russel, A. (2004): Annotating multi-media / multi-modal resources with elan. In: Lino, M., Xavier, M., Ferreire, F., Costa, R., Silva, R. (eds.) Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC). Lisbon: Portugal, 2065–2068. (ELAN 4.5.1 software can be downloaded at http://tla.mpi.nl/tools/tla-tools/elan/download/)

Bunt, H., Alexandersson, J., Choe, J.W., Fang, A.Ch., Hasida, K., Petukhova, V., Popescu-Belis, A. & Traum D. (2012): ISO 24617-2: A semantically-based standard for dialogue annotation. Proceedings of LREC, 8th International Conference on Language Resources and Evaluation, 430-437.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D. & Wellner. P. (2005): The AMI Meeting Corpus: A Pre-Announcement. Proceedings of the Second international conference on Machine Learning for Multimodal Interaction, 28–39.

Ekman, P. & Friesen (1969): The repertoire of non-verbal behavior: Categories, origins, usage and coding. Semiotica 1(1): 49–98.

Ekman, P. & Friesen, W. V. (1978): FACS- Facial Action Coding System. Carnegie Mellon School of Computer Science.

Ekman, P., Friesen, W. V. & Hager, J. C. (2002): Facial Action Coding System: The Manual on CD ROM. A Human Face, Salt Lake City. (Description of FACS and information on the CD ROM is available at: http://face-and-emotion.com/dataface/facs/description.jsp)

Ekman, P. & Rosenberg, E.L. (1998): What the Face Reveals. Oxford: Oxford University Press.

Enfield, N.J. (2009): The Anatomy of Meaning. Speech, gesture, and composite utterances. Cambridge: Cambridge University Press.

Evans, J.L., Alibali, M.W. and McNeill, N.M. (2001): Divergence of verbal expression and embodied knowledge: Evidence from speech and gesture in children with specific language impairment. Language and Cognitive Processes 16(2-3), 309–331.

Foster, M.E. and Oberlander, J. (2007): Corpus-based generation of head and eyebrow motion for an embodied conversational agent. Language Resources and Evaluation Conference (LREC) 41 (3/4), 305–323.

Hunyadi, L., Földesi, A., Szekrényes, I., Staudt, A, Kiss, H., Abuczki, A. & Bódog A. (2012a): Az ember-gép kommunikáció elméleti-technológiai modellje és nyelvtechnológiai vonatkozásai. In: Kenesei, I., Prószéky, G. & Várady T. (eds.): Általános Nyelvészeti Tanulmányok XXIV. Nyelvtechnológiai kutatások. Budapest: Akadémiai Kiadó, 265–309.

Hunyadi, L., Szekrényes, I., Borbély, A. & Kiss H. (2012b): Annotation of spoken syntax in relation to prosody and multimodal pragmatics. Proceedings of IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom), 537–541.

Kendon, A. (2004): Gesture. Visible Action as Utterance. Cambridge: CUP.

Knight, D., Bayoumi, S., Mills, S., Crabtree, A., Adolphs, S., Pridmore, T. & Carter, R.A. (2006): Beyond the Text: Construction and Analysis of Multi-Modal Linguistic Corpora. Proceedings of the 2nd International Conference on e-Social Science, Manchester, 28–30 June 2006 [online]. Available at: http://www.ncess.ac.uk/events/conference/2006/papers/abstracts/KnightBeyondTheText.shtml

Knight, D, (2009): A Multimodal Corpus Approach to the Analysis of Backchannelling Behaviour (doctoral dissertation). The University of Nottingham.

Massaro, D. W. (1987): Speech Perception by Ear and Eye: A Paradigm for Psychological Inquiry. Hillsdale, New Jersey: Lawrence Erlbaum.

McCowan, I., Carletta, J., Kraaij, W., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., Post, W., Reidsma, D. & Wellner. P. (2005): The AMI Meeting Corpus. Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research.

McNeill, D. (1992): Hand and Mind: What Gestures Reveal about Thought. Chicago: University Of Chicago Press.

Pápay, K., Szeghalmy, S. & Szekrényes, I. (2011): HuComTech Multimodal Corpus Annotation. Argumentum 7, 330–347.

Saferstein, B. (2004): Digital technology- methodological adoption: Text and video as a resource for analytical reflectivity. Journal of Applied Linguistics, 1(2), 197–223.

Schiel, F., Steininger, S. & Türk, U. (2002): The SmartKom Multimodal Corpus at BAS. Proceeding of the 3rd Language Resources & Evaluation Conference (LREC).

Steininger, S., Rabold, S., Dioubina, O. & Schiel, F. (2002): Development of the User-State Conventions for the Multimodal Corpus in SmartKom. SmartKom. LREC Workshop Multimodal Resources.

Vilhjalmsson H, Cantelmo N, Cassell J, Chafai NE, Kipp M, & Kopp S (2007): The Behavior Markup Language: Recent Developments and Challenges. In: Proc. of Intelligent Virtual Agents (IVA 2007). LNAI, 4722. Berlin, Heidelberg: Springer, 99–111.

Annotation procedures, feature extraction and query options

Ágnes Abuczki

The present chapter provides guidelines on the usability of two annotation tools, ELAN and Praat, in conversation analysis, in particular, in the analysis of discourse marker use in spontaneous dialogues.

1) Annotation procedures

As for the annotation of the corpus material, different annotation tools have been used to annotate the audio and video contents of a corpus that can later be merged in query systems (ELAN) or databases (MySQL). The HuComTech corpus is annotated at multiple multimodal levels in Praat, a fine grained audio analysis tool (Boersma & Weenink 2007) for the audio material and Quannot, a custom designed environment for the video material). Praat (Boersma & Weenink 2007) enables a much more precise and detailed acoustic analysis than compact multimodal annotation software such as Anvil (Multimodal Annotation and Visualization Tool[1)] or ELAN (Brugman & Russel 2004: 2065–2068). First, annotators were trained to follow the Annotation guidelines of the HuComTech corpus handed in test annotations and discussed them in regular annotators' meetings.

First of all, their task was to transcribe the recordings and manually align the phrases to the speech signal in Praat. Verbatim (word-for-word) transcriptions involve non-lexical/nonverbal vocalisations, such as filled pauses, hesitations, breathing, coughing, sneezing, lip smack, and non-speaker voices, such as beep sound (used for the synchronization procedure of the audio and video material) as well. At the discourse level of audio annotation, the transcribed dialogue is segmented into floor control (also called turn segment) types (turn-take, turn-keep, turn-give, backchannel). In the Annotation guidelines, a turn is defined as a sequence of speech uttered by a single speaker, surrounded by silence and/or followed by speaker change. Acoustic features (durational data, pitch and intensity values, pitch movement and silence detection) were automatically extracted using a Praat script (Boersma & Weenink 2007) because these prosodic features may mark how an utterance is meant; therefore, they are necessary for the analysis and functional disambiguation of DMs and their host units. The pitch movement of the speakers was also automatically annotated using a Prosogram-based application further developed by the Computational Linguistics Subproject of the HuComTech

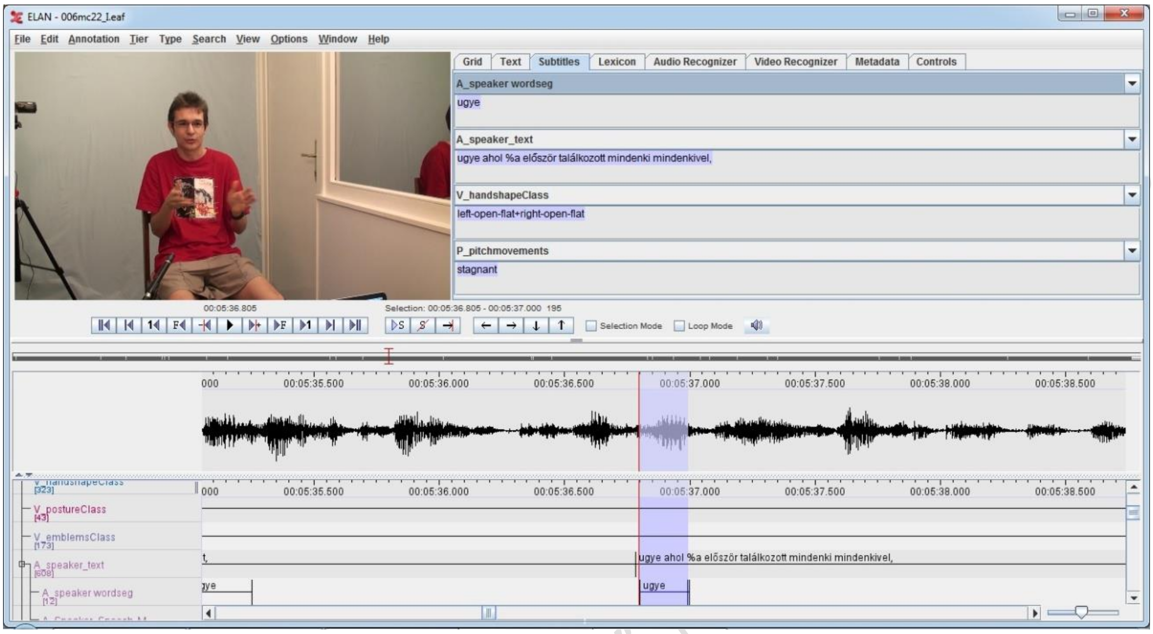
[1] ANVIL is freely available at: http://www.anvil-software.org/

project (Szekrényes, Csipkés & Oravecz 2011, Hunyadi, Szekrényes, Borbély & Kiss 2012). We found it crucial to annotate intonation since it implicitly express the emotional state and attitudes of the speaker, add subtlety to meaning and may help distinguish various functions.

On the other hand, the video contents of the HuComTech corpus were annotated in Qannot (Pápay, Szeghalmy & Szekrényes 2011: 330–347, Hunyadi, Földesi, Szekrényes, Staudt, Kiss, Abuczki, & Bódog 2012: 265–309) which was designed instead of Anvil because Anvil sometimes failed to handle large files, and there was a risk that annotations might be out of snyc in these large files. The video annotation of the corpus involves the labelling of facial expressions, gaze, eyebrows, head movement, hand shape, posture, touch motion, deictic gestures and emblems. Within this paper, three levels of the video annotation will be analysed in relation to DM segments: (1) gaze direction, (2) hand movements, ad (3) facial expressions. Gaze direction is analysed owing to its several functions it plays in discourse. It is used to collect information in our environment, to express our emotions, attitudes and intentions, thereby influencing our conversation partners. Eyes reflect the degree of being interested and involved in conversation, and different gaze directions express our intention to take, keep or give the floor over to the next speaker. Gaze may also serve as a deictic device by looking at the desired object one want to get. Secondly, hand movements are also indispensable to analyse in relation to DMs since manual gestures include - among others - deictic, iconic and discourse structuring gestures which all reveal important aspects of discourse and contribute to the correct interpretation of utterances.

When the annotations were completed, the various annotation files of the HuComTech corpus were merged in an SQL database. Annotations are still stored in SQL and can also be queried in a very user-friendly way using the ELAN software (Brugman & Russel 2004). Custom query options of ELAN include: *N-gram within annotations; Structured search of multiple files; Multiple layer search; Find overlapping labels within a file;* and *Find left overlaps within a file*, etc. The availability of multimodal annotation tiers enables the systematic and joint search of the temporal alignment and/or synchronous co-occurrences of turns, clauses or specific lexical items with the use of manual gestures, head movement types, gaze directions, eyebrow movement types and posture changes in spontaneous interaction corpora.

The methodology used involve (1) segmentation of DMs (shown in Figure 1); (2) analysis of their further textual, acoustic and visual features (position of the DM within the turn/utterance, sequential properties, statistical analysis of co-occurrence patterns) by using the custom query options of ELAN (Brugman & Russel 2004) such as *N-gram within annotations; Structured search of multiple files; Find overlapping labels within a file;* and *Find left overlaps within a file*; (3) silence annotation in Praat (Boersma & Weenink 2007), and then measuring the segment duration of the host unit and its preceding silence; as well as (4) collecting the synchronously performed hand gestures. Therefore, vertical label statistics have also been performed in order to reveal the co-occurrence of the audio labels with labels from the visual domain such as hand shape types.



A_speaker wordseg: ugye

A_speaker_text: ugye ahol %a először találkozott mindenki mindenkivel,

V_handshapeClass: left-open-flat+right-open-flat

P_pitchmovements: stagnant

Selection: 00:05:36.805 - 00:05:37.000 195

Figure 1 *DM segmentation in ELAN*

2) Feature extraction procedures after segmenting DMs

First, I briefly describe the mechanism of automatic prosodic annotation, then I overview the query options I used in ELAN during the analysis of DMs.

There are various techniques available to implement the automatic annotation of intonation. One of the most well-known and widely used annotation conventions is the tones and break indices labelling standard (ToBi) (Rosenberg 2010: 146–150) which is suitable for capturing prosodic structure by labelling pitch levels. The main problem with ToBi is that it has not been adapted to Hungarian yet. Conversely, the Tilt intonation model (Taylor 1998) is language independent. The system operates the basic unit of intonational events which are described as movements based on the RCF (rise/fall/connection) model. Tilt uses a set of continuous parameters determined from the examination of the local shape of the F0 contour to label pitch movements. However, the goal of the HuComTech project is to examine and annotate the movements of F0 contour but not in this parametric manner; instead, they use a different stylization, trend calculation, segmentation, and classification methods. Since they want to illustrate the perceived intonation characterized by long stretches of smooth F0-movements rather than frequent significant momentary excursions of the actual measured F0 data, stylization of F0 curves is performed on the extracted pitch data (Hunyadi, Szekrényes, Borbély & Kiss 2012). There is an existing stylization method available which is theoretically founded and has a widely-tested application called Prosogram (d'Alessandro & Mertens 2004, Mertens 2004) The theoretical assumption underlying the application is the psychoacoustic model of tonal perception which concentrates on the perceptually relevant features of intonation, distinguishing audible F0 variations from inaudible ones (t'Hart 1976: 11–19). The Prosogram application (Hunyadi, Szekrényes, Borbély & Kiss 2012) was further developed within the speech processing subproject of the HuComTech research group and was implemented as a Praat script for automatic F0 stylization. For customization to our corpus and goals, Hunyadi, Szekrényes, Borbély & Kiss (2012) have developed a new script to make F0 stylizations for the recordings in HuComTech database sentence by sentence (with sentence boundaries manually annotated in syntactic annotation) using external pitch data generated before. Extremely high or low values were dropped from the stylization using the 10th and the 90th percentile value as a threshold. The Prosogram stores stylization in PitchTier files as well, and later the textual transcriptions of the recordings are also added to the graphical output. They (Hunyadi, Szekrényes, Borbély & Kiss 2012) use dynamic, speaker-dependent pitch range to extract F0 data and use a stylization method to calculate more holistic trend-lines from stylizations to describe the movement of pitch. Along trend-lines (point to point) pitch movement can be segmented into blocks for labelling. The standard deviation of the pitch values is used as a threshold to classify the movement progress using five simple categories: *rise*, *fall*, *upward*, *descending* and *stagnant* (see Figure 2).

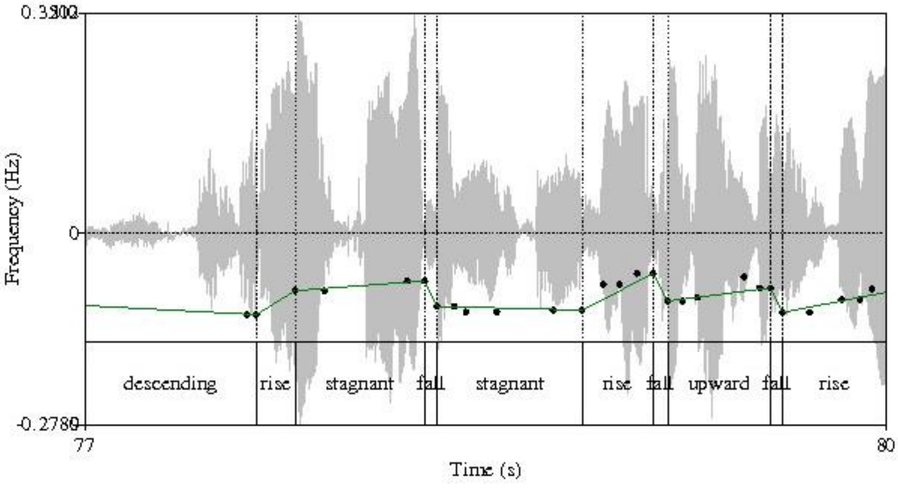
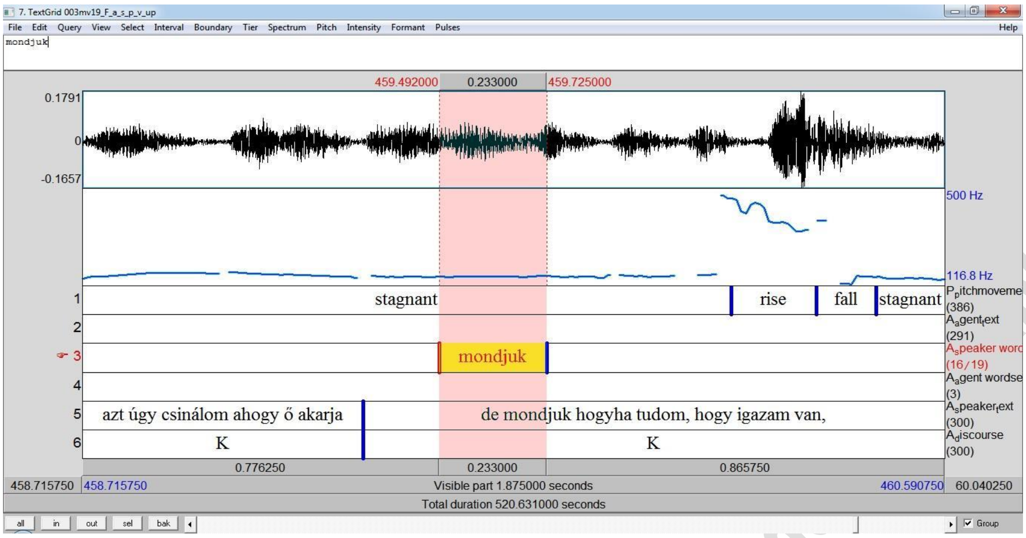


Figure 2 *Segmentation and classification of pitch movements using ProsoTool* (source: Szekrényes to appear)

However, to automatically perform prosodic annotation, I first had to export my .eaf annotation files from .eaf file format (from ELAN) into Praat TextGrids. You can simultaneously export multiple files as Praat TextGrids, so the procedure of this conversion can be instantly performed.

After annotating sound files, Prosogram also stores stylization in PitchTier files; therefore, they can be viewed in Praat as well. The following figure shows the result of exporting both automatic ProsoTool annotation results and manual DM segment annotation .eaf files into Praat TextGrids (Figure 3).


7. TextGrid 003mv19_F_a_s_p_v_up
File Edit Query View Select Interval Boundary Tier Spectrum Pitch Intensity Formant Pulses Help
mondjuk
459.492000 0.233000 459.725000
0.1791
0
-0.1657
500 Hz
116.8 Hz
1 stagnant rise fall stagnant
2
3 mondjuk
4
5
6 K K
Ppitchmoveme (386)
Aagenttext (291)
Aspeaker word (16/19)
Aagent wordse (3)
Aspeakertext (300)
Adiscourse (300)
0.776250 0.233000 0.865750
458.715750 458.715750 Visible part 1.875000 seconds 460.590750 60.040250
Total duration 520.631000 seconds
all in out sel bak
Group

azt úgy csinálom ahogy ő akarja

de mondjuk hogyha tudom, hogy igazam van,

Figure 3 *ProsoTool results and ELAN DM segment annotations both exported into Praat*

Finally, PitchTier Files can be exported into several other software, such as ELAN, for further queries.

3) Automatic annotation into sounding and silent parts

A further acoustic feature whose role in discourse interpretation is analysed in this paper is silence. This paragraph briefly describes the process of automatic silence annotation in Praat and its query results in ELAN (Brugman & Russel 2004). This analysis followed the segmentation of DMs and it was performed with the aim to test if DMs are predominantly preceded by silence (as they are often described in the literature, such as in (Fraser 1990)). The phonetic parameters set for automatic silence annotation were as follows:

- minimum pitch (to measure intensity): 100 Hz (subtract mean)
- time step: automatic (0,01 s)
- silence threshold: - 45 dB
- minimum silent interval duration: 0,15 s
- minimum sounding interval duration: 0,05 s

As a result, annotation segmented the recordings into sounding and silent segments (see Figure 4).

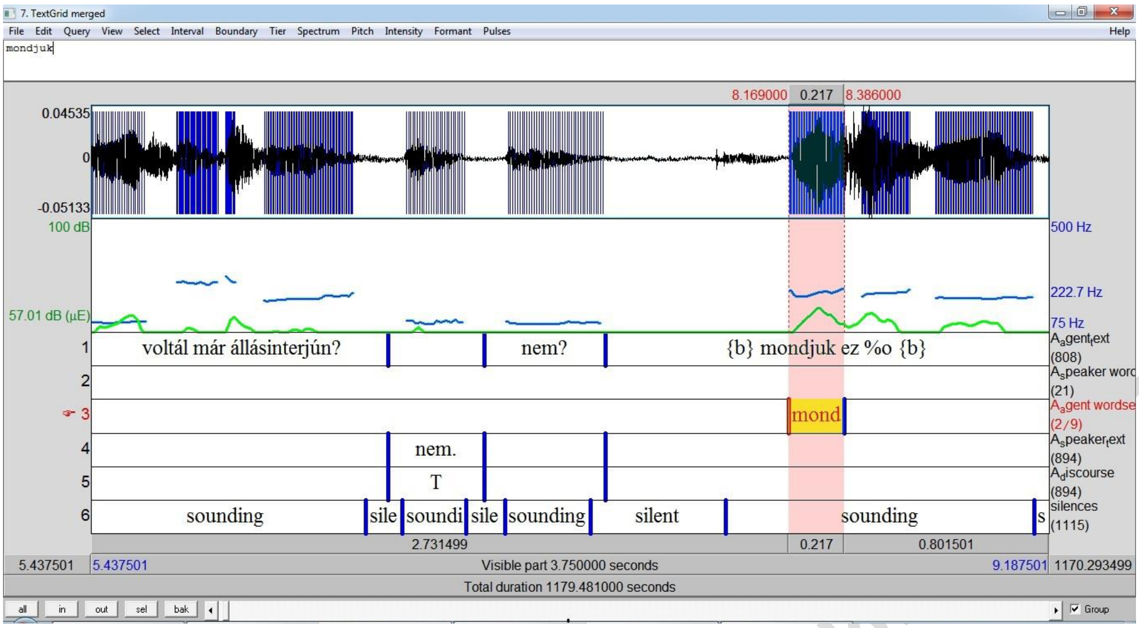
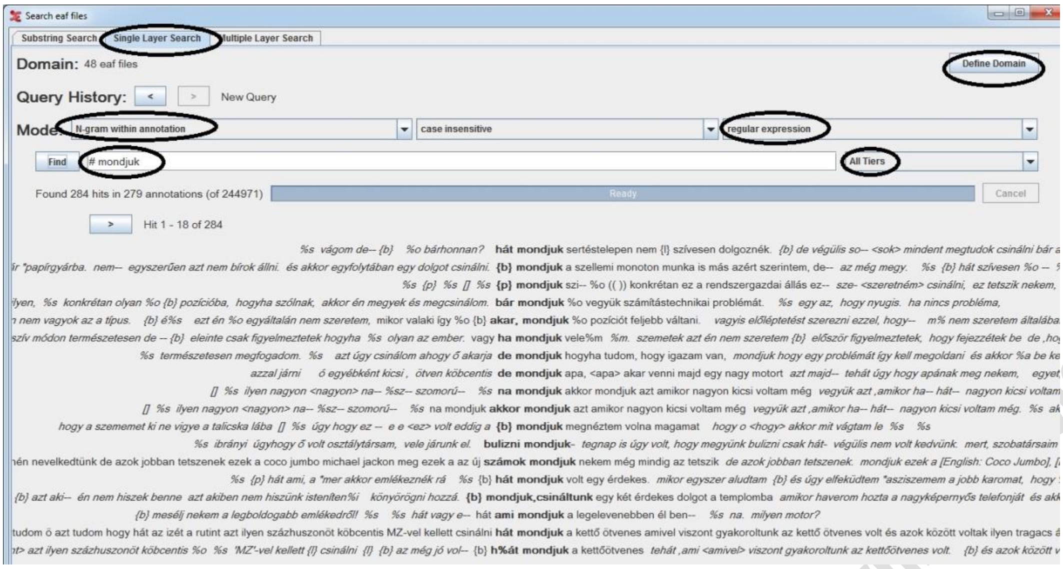

7. TextGrid merged
File Edit Query View Select Interval Boundary Tier Spectrum Pitch Intensity Formant Pulses Help
mondjuk
8.169000 0.217 8.386000
0.04535
0
-0.05133
100 dB
57.01 dB (µE)
500 Hz
222.7 Hz
75 Hz
1 voltál már állásinterjún? nem? {b} mondjuk ez %o {b}
2
3 mond
4 nem.
5 T
6 sounding sile soundi sile sounding silent sounding s
Agentext (808)
Aspeaker word (21)
Agent wordse (2/9)
Aspeakerext (894)
Adiscourse (894)
silences (1115)
2.731499 0.217 0.801501
5.437501 5.437501 Visible part 3.750000 seconds 9.187501 1170.293499
Total duration 1179.481000 seconds
all in out sel bak
Group

Figure 4 *Automatic annotation into sounding and silent segments in Praat*

Afterwards, labelling was merged with existing annotations and exported into ELAN (Brugman & Russel 2004) for querying. Silence annotation is useful when we want to empirically investigate if DMs are surrounded by silent parts or not. Of course, the duration of silence might be set to other parameters as well. In this study I wanted to consider relatively short silences as well, that is why I set the minimum silent interval duration as 150 ms (since it is audible).

4) Query options in ELAN

When all manual and automatic annotations are merged in an .eaf file, possible query options in ELAN include various *Single Layer Search* and *Multiple Layer Search* options. Options *of Single Layer Search* include for instance '*Search for N-gram within annotations'* (see Figure 5) which helps us identify the co-occurrence patterns of the items. You can search the environment of the selected segment in the same tier (in our case, in the text tier): what it is left context (using *# mondjuk*), what it is right context (using *mondjuk #).* In Figure 5 I searched in the search domain of 48 annotation files for all instances where *mondjuk* (either in the textual transcription of the agent's or speaker's speech) is preceded by something (that is, it is not in a segment-initial position). First I had to set the mode from '*exact search'* to '*regular expressions'* because I used the # regexp in the search box. In this example, I wanted to search both the agent's and speakers' text tiers, therefore, I chose to search all tiers.



Domain: 48 eaf files

Query History: < > New Query

Mode: N-gram within annotation | case insensitive | regular expression

mondjuk | All Tiers

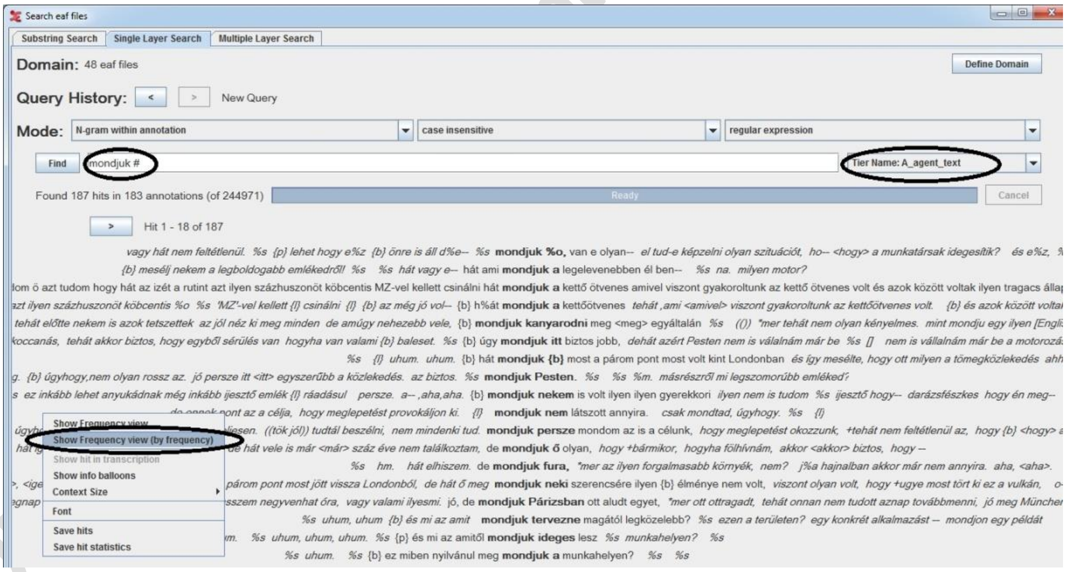
Found 284 hits in 279 annotations (of 244971)

Hit 1 - 18 of 284

```
                                        %s  vágom de-- {b}   %o bárhonnan?  hát mondjuk sertéstelepen nem {l} szívesen dolgoznék.  {b} de végülis so-- <sok> mindent megtudok csinálni bár a
ár *papírgyárba.  nem--  egyszerűen azt nem bírok állni.  és akkor egyfolytában egy dolgot csinálni.  {b} mondjuk a szellemi monoton munka is más azért szerintem, de--  az még megy.   %s  {b} hát szívesen %o --  %
                                                                      %s  {p}  %s  [] %s  {p} mondjuk szi-- %o (( )) konkrétan ez a rendszergazdai állás ez--  sze- <szeretném> csinálni,  ez tetszik nekem,
ilyen,  %s  konkrétan olyan %o {b} pozícióba,  hogyha szólnak,  akkor én megyek és megcsinálom.  bár mondjuk %o vegyük számítástechnikai problémát.   %s  egy az,  hogy nyugis.  ha nincs probléma,
n nem vagyok az a típus.   {b} é%s   ezt én %o egyáltalán nem szeretem,  mikor valaki így %o {b} akar, mondjuk %o pozíciót feljebb váltani.   vagyis előléptetést szerezni ezzel, hogy--   m% nem szeretem általába
szív módon természetesen de -- {b}  eleinte csak figyelmeztetek hogyha  %s  olyan az ember.  vagy ha mondjuk vele%m  %m.  szemetek azt én nem szeretem {b}  először figyelmeztetek,  hogy fejezzétek be  de ,ho
                        %s  természetesen megfogadom.  %s   azt úgy csinálom ahogy ő akarja  de mondjuk hogyha tudom, hogy igazam van,  mondjuk hogy egy problémát így kell megoldani  és akkor %a be ke
                                            azzal járni     ő egyébként kicsi ,  ötven köbcentis  de mondjuk apa, <apa> akar venni majd egy nagy motort  azt majd--  tehát úgy hogy apának meg nekem,   egyet
                                 [] %s  ilyen nagyon <nagyon> na-- %sz-- szomorú--   %s  na mondjuk akkor mondjuk azt amikor nagyon kicsi voltam még  vegyük azt ,amikor ha-- hát--  nagyon kicsi voltam
                       [] %s  ilyen nagyon <nagyon> na-- %sz-- szomorú--   %s  na mondjuk akkor mondjuk azt amikor nagyon kicsi voltam még  vegyük azt ,amikor ha-- hát--  nagyon kicsi voltam még.  %s  ak
          hogy a szememet ki ne vigye a talicska lába  [] %s  úgy hogy ez --  e e <ez> volt eddig a  {b} mondjuk megnéztem volna magamat   hogy o <hogy> akkor mit vágtam le  %s   %s
                                  %s  ibrányi  úgyhogy ő volt osztálytársam,  vele járunk el.   bulizni mondjuk-  tegnap is úgy volt, hogy megyünk bulizni csak hát-  végülis nem volt kedvünk.  mert, szobatársaim
nén nevelkedtünk de azok jobban tetszenek ezek a coco jumbo michael jackon meg ezek a az új számok mondjuk nekem még mindig az tetszik  de azok jobban tetszenek.  mondjuk ezek a [English: Coco Jumbo], [
                                         %s  {p} hát ami, a *mer akkor emlékezném rá   %s {b} hát mondjuk volt egy érdekes.  mikor egyszer aludtam  {b} és úgy elfeküdtem *asziszemem a jobb karomat,  hogy
 {b} azt aki--  én nem hiszek benne  azt akiben nem hiszünk isteníten%i   könyörögni hozzá.  {b} mondjuk,csináltunk egy két érdekes dolgot a templomba  amikor haverom hozta a nagyképernyős telefonját  és akk
                       {b} mesélj nekem a legboldogabb emlékedről!  %s   %s  hát vagy e-- hát ami mondjuk a legelevenebben él ben--   %s  na.  milyen motor?
tudom ö azt tudom hogy hát az izét a rutint azt ilyen százhuszonöt köbcentis MZ-vel kellett csinálni hát mondjuk a kettő ötvenes amivel viszont gyakoroltunk az kettő ötvenes volt és azok között voltak ilyen tragacs á
nt> azt ilyen százhuszonöt köbcentis %o  %s  'MZ'-vel kellett {l} csinálni  {l}  {b} az még jó vol-- {b} h%át mondjuk a kettőötvenes  tehát ,ami <amivel> viszont gyakoroltunk az kettőötvenes volt.   {b} és azok között v
```

Figure 5 *Concordance view of the search ‘# mondjuk’*

The software also enables the researcher to search the transcription of only one of the speakers. To do so, you simply have to switch the search mode from ‘All Tiers’ to ‘A_agent_text’ and change the search term to ‘*mondjuk #’*. For example, Figure 6 shows the concordance view of a search to find out what elements *mondjuk* is followed by in the interviewer’s speech.



Domain: 48 eaf files

Query History: < > New Query

Mode: N-gram within annotation | case insensitive | regular expression

mondjuk # | Tier Name: A_agent_text

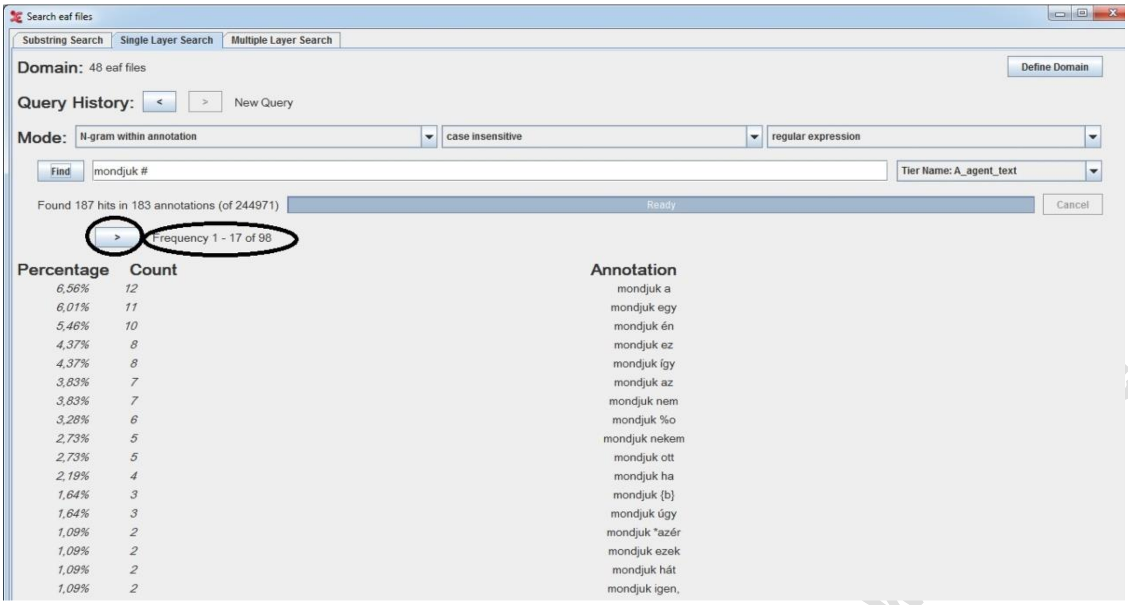
Found 187 hits in 183 annotations (of 244971)

Hit 1 - 18 of 187

```
                   vagy hát nem feltétlenül.  %s  {p} lehet hogy e%z  {b} önre is áll d%e--  %s  mondjuk %o, van e olyan--  el tud-e képzelni olyan szituációt,  ho-- <hogy> a munkatársak idegesítik?   és e%z,  %
                   {b} mesélj nekem a legboldogabb emlékedről!  %s   %s  hát vagy e--  hát ami mondjuk a legelevenebben él ben--   %s  na.  milyen motor?
lom ö azt tudom hogy hát az izét a rutint azt ilyen százhuszonöt köbcentis MZ-vel kellett csinálni hát mondjuk a kettő ötvenes amivel viszont gyakoroltunk az kettő ötvenes volt és azok között voltak ilyen tragacs álla
azt ilyen százhuszonöt köbcentis %o  %s  'MZ'-vel kellett {l} csinálni  {l}  {b} az még jó vol-- {b} h%át mondjuk a kettőötvenes  tehát ,ami <amivel> viszont gyakoroltunk az kettőötvenes volt.   {b} és azok között volta
 tehát előtte nekem is azok tetszettek  az jól néz ki meg minden  de amúgy nehezebb vele,  {b} mondjuk kanyarodni meg <meg> egyáltalán  %s   (()) *mer tehát nem olyan kényelmes.  mint mondju egy ilyen [Engli
koccanás,  tehát akkor biztos, hogy egyből sérülés van  hogyha van valami {b} baleset.  %s  {b} úgy mondjuk itt biztos jobb,  dehát azért Pesten nem is válalnám már be  %s  []   nem is vállalnám már be a motorozá
                                                           %s   {l}  uhum.  uhum.  {b} hát mondjuk {b} most a párom pont most volt kint Londonban  és így mesélte, hogy ott milyen a tömegközlekedés  ahh
g.  {b} úgyhogy,nem olyan rossz az.  jó persze itt <itt> egyszerűbb a közlekedés.  az biztos.  %s  mondjuk Pesten.  %s   %s  %m.  másrészről mi legszomorúbb emléked?
s  ez inkább lehet anyukádnak még inkább ijesztő emlék {l} ráadásul   persze.  a-- ,aha,aha.  {b} mondjuk nekem is volt ilyen ilyen gyerekkori  ilyen nem is tudom  %s  ijesztő hogy--  darázsfészkes  hogy én meg--
                        de ennek pont az a célja,  hogy meglepetést provokáljon ki.   {l}   mondjuk nem látszott annyira.   csak mondtad, úgyhogy.  %s   {l}
úgyh                         iesen.  ((tök jól)) tudtál beszélni,  nem mindenki tud.  mondjuk persze mondom az is a célunk,  hogy meglepetést okozzunk,  +tehát nem feltétlenül az,  hogy {b} <hogy> a
hát ig                       de hát vele is már <már> száz éve nem találkoztam,  de mondjuk ő olyan,  hogy +bármikor,  hogyha fölhívnám,  akkor <akkor> biztos,  hogy --
                                                            %s   hm.   hát elhiszem.  de mondjuk fura,  *mer az ilyen forgalmasabb környék,  nem?   j%a hajnalban akkor már nem annyira.  aha, <aha>.
>, <ige                      párom pont most jött vissza Londonból,  de hát ő meg  mondjuk neki szerencsére ilyen {b} élménye nem volt,  viszont olyan volt,  hogy +ugye most tört ki ez a vulkán,   o-
egnap                        sszem negyvenhat óra,  vagy valami ilyesmi.  jó, de mondjuk Párizsban ott aludt egyet,  *mer ott ottragadt,  tehát onnan nem tudott aznap továbbmenni,  jó meg Müncher
                                                %s  uhum, uhum  {b} és mi az amit   mondjuk tervezne magától legközelebb?  %s  ezen a területen?  egy konkrét alkalmazást --  mondjon egy példát
                             m.   %s  uhum, uhum, uhum.  %s  {p} és mi az amitől mondjuk ideges lesz  %s  munkahelyen?   %s
                                                 %s  uhum.   %s  {b} ez miben nyilvánul meg mondjuk a munkahelyen?   %s   %s
```

Figure 6 *Concordance view of the search results of ‘mondjuk #’*

To do so, switch from Concordance view to Frequency view by right-clicking and choosing ‘*Show Frequency view (by frequency)*’ from the drop-down menu. Figure 7 shows the first page of the results of this frequency query. You can jump to the following search result pages by clicking on the > button. You can move among the several queries and result pages by clicking on either the < (next) or > (previous) buttons.



Domain: 48 eaf files

Query History: New Query

Mode: N-gram within annotation | case insensitive | regular expression

mondjuk #

Tier Name: A_agent_text

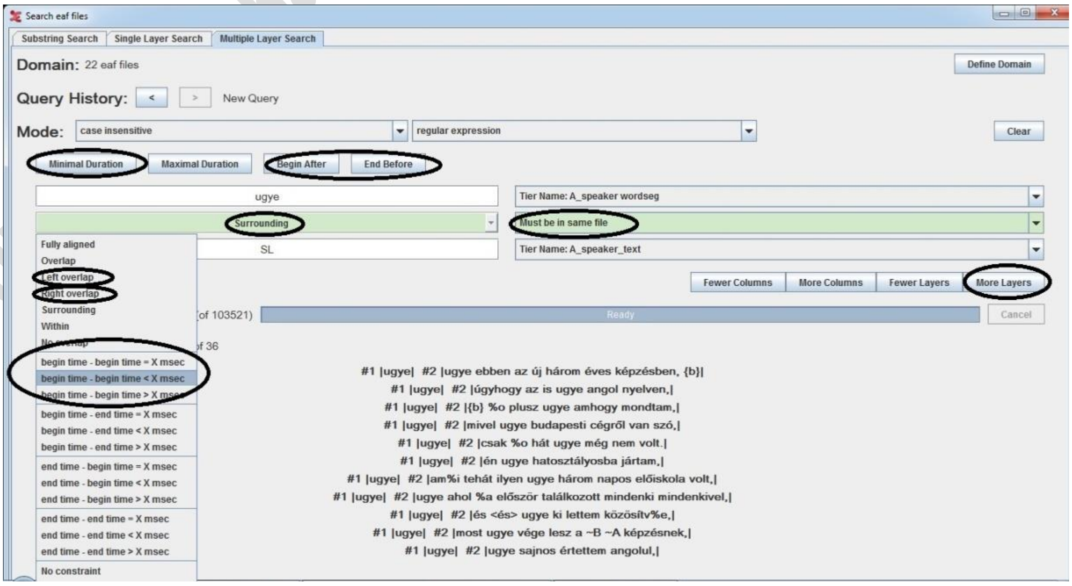
Found 187 hits in 183 annotations (of 244971)

Frequency 1 - 17 of 98

Percentage	Count	Annotation
6,56%	12	mondjuk a
6,01%	11	mondjuk egy
5,46%	10	mondjuk én
4,37%	8	mondjuk ez
4,37%	8	mondjuk így
3,83%	7	mondjuk az
3,83%	7	mondjuk nem
3,28%	6	mondjuk %o
2,73%	5	mondjuk nekem
2,73%	5	mondjuk ott
2,19%	4	mondjuk ha
1,64%	3	mondjuk {b}
1,64%	3	mondjuk úgy
1,09%	2	mondjuk *azér
1,09%	2	mondjuk ezek
1,09%	2	mondjuk hát
1,09%	2	mondjuk igen,

Figure 7 *Frequency view of the search 'mondjuk #' by frequency in decreasing order*

One can also search what labels in other tiers a segment (e.g. *ugye* wordseg) entirely or partially overlaps with (left or right overlap). For this purpose, you need to use Multiple Layer Search and have to set the names of the labels and tiers to be searched. For instance, if you want to find out if *ugye* is a separate unit and phonologically independent or not, you can choose from several search options: (1) search for instances when ugye is surrounded by SL (silence) (shown in Figure 8), (2) search for either left or right overlaps of *ugye* with SL (this will show instances when *ugye* is either preceded or followed by silence), (3) search using custom-defined temporal constraints. The most exact results you can achieve using the third method, that is, if you set the time difference allowed between the begin time of a segment (e.g. *ugye* in wordseg tier) and another one (e.g. *SL* in text tier).


Search eaf files
Substring Search
Single Layer Search
Multiple Layer Search
Define Domain
<
>
Clear
Minimal Duration
Maximal Duration
Begin After
End Before
Fully aligned
Overlap
Left overlap
Right overlap
Surrounding
Within
No overlap
begin time - begin time = X msec
begin time - begin time < X msec
begin time - begin time > X msec
begin time - end time = X msec
begin time - end time < X msec
begin time - end time > X msec
end time - begin time = X msec
end time - begin time < X msec
end time - begin time > X msec
end time - end time = X msec
end time - end time < X msec
end time - end time > X msec
No constraint
Fewer Columns
More Columns
Fewer Layers
More Layers
Ready
Cancel

Domain: 22 eaf files

Query History: New Query

Mode: case insensitive | regular expression

ugye | Tier Name: A_speaker wordseg

Surrounding | Must be in same file

SL | Tier Name: A_speaker_text

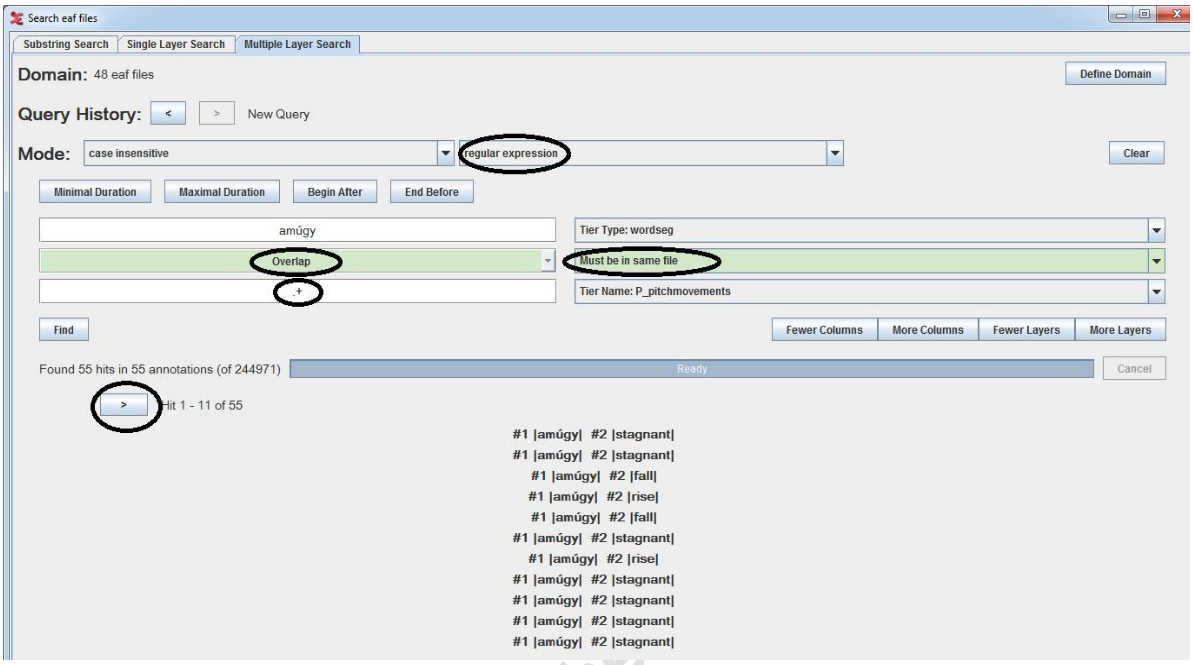
(of 103521)

of 36

#1 |ugye| #2 |ugye ebben az új három éves képzésben, {b}|
#1 |ugye| #2 |úgyhogy az is ugye angol nyelven,|
#1 |ugye| #2 |{b} %o plusz ugye amhogy mondtam,|
#1 |ugye| #2 |mivel ugye budapesti cégről van szó,|
#1 |ugye| #2 |csak %o hát ugye még nem volt.|
#1 |ugye| #2 |én ugye hatosztályosba jártam,|
#1 |ugye| #2 |am%i tehát ilyen ugye három napos előiskola volt,|
#1 |ugye| #2 |ugye ahol %a először találkozott mindenki mindenkivel,|
#1 |ugye| #2 |és <és> ugye ki lettem közösítv%e,|
#1 |ugye| #2 |most ugye vége lesz a ~B ~A képzésnek,|
#1 |ugye| #2 |ugye sajnos értettem angolul,|

Figure 8 *Search for instances when* ugye *is preceded by silence*

As shown in Figure 9, there is an option to search for labels (e.g. pitch movement, position or thematic labels) overlapping with the target word segment. If you do not look for a certain pitch movement, but rather would like to search the distribution of all pitch movement types, you may use the regular expression '.+' which looks for all labels in the specified tier.



Domain: 48 eaf files

Query History: New Query

Mode: case insensitive | regular expression

amúgy | Tier Type: wordseg

Overlap | Must be in same file

.+ | Tier Name: P_pitchmovements

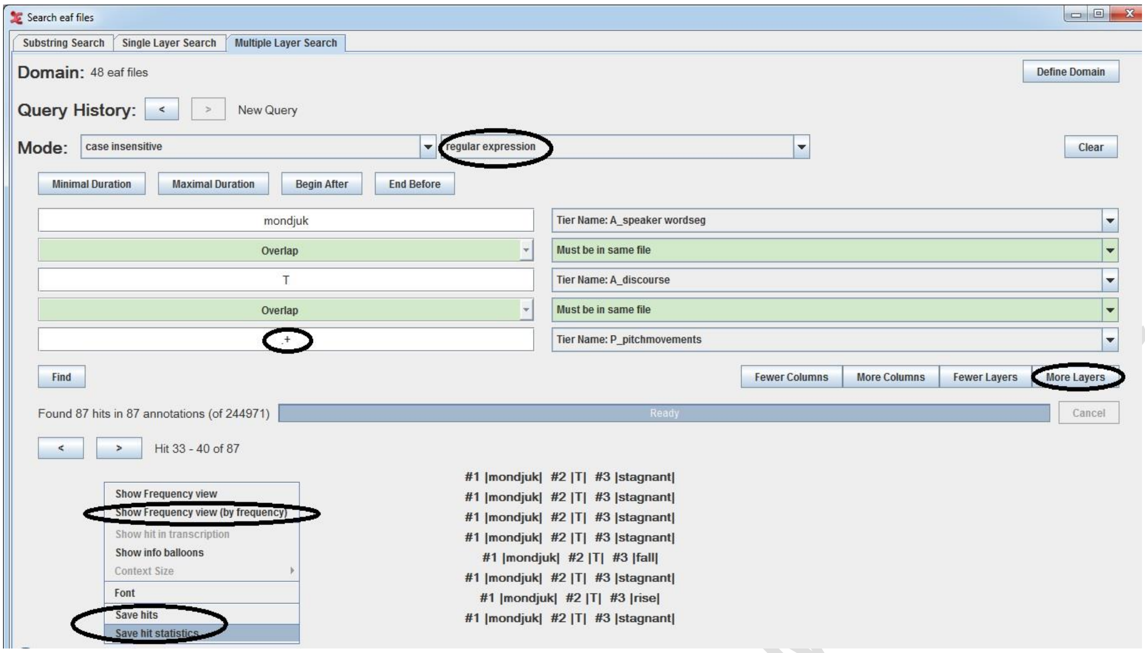
Found 55 hits in 55 annotations (of 244971)

Hit 1 - 11 of 55

```
#1 |amúgy|  #2 |stagnant|
#1 |amúgy|  #2 |stagnant|
#1 |amúgy|  #2 |fall|
#1 |amúgy|  #2 |rise|
#1 |amúgy|  #2 |fall|
#1 |amúgy|  #2 |stagnant|
#1 |amúgy|  #2 |rise|
#1 |amúgy|  #2 |stagnant|
#1 |amúgy|  #2 |stagnant|
#1 |amúgy|  #2 |stagnant|
#1 |amúgy|  #2 |stagnant|
```

Figure 9 *First page of the search results for pitch movement overlapping with the wordseg amúgy*

If you want to refine your search and add more constraints, you may add more columns or tiers to advance your search. For instance, Figure 10 shows a search for the pitch movement types ('.+' stands for any type/any label/any word) of only those *mondjuk* word segments which are in turn-initial (utterance-initial) positions (marked by T as turn-take in the audio annotation of the HuComtech corpus). If you want to get the results in frequency view, you can achieve this by right-clicking on the results and choosing this option. Finally, queries in ELAN can be saved as .xls files (by clicking on 'Save hits' or 'Save hit statistics'), which enables us to perform calculations on them.



Domain: 48 eaf files

Mode: case insensitive | regular expression

mondjuk	Tier Name: A_speaker wordseg
Overlap	Must be in same file
T	Tier Name: A_discourse
Overlap	Must be in same file
+	Tier Name: P_pitchmovements

Found 87 hits in 87 annotations (of 244971)

Hit 33 - 40 of 87

```
#1 |mondjuk|  #2 |T|  #3 |stagnant|
#1 |mondjuk|  #2 |T|  #3 |stagnant|
#1 |mondjuk|  #2 |T|  #3 |stagnant|
#1 |mondjuk|  #2 |T|  #3 |stagnant|
   #1 |mondjuk|  #2 |T|  #3 |fall|
#1 |mondjuk|  #2 |T|  #3 |stagnant|
   #1 |mondjuk|  #2 |T|  #3 |rise|
#1 |mondjuk|  #2 |T|  #3 |stagnant|
```

Figure 10 *An example for three-layered multiple layer search*

After the queries, the following statistical tests were performed on the data in SPSS 19.0[2]: descriptive and inferential statistical tests, including Pearson's chi-square test, Fischer's exact test, Crosstabs test, independent samples t-test, paired t-test, and drawing box plot graphs. Descriptive statistical tests simply measured the frequency of the use of the selected items based on gender, speaker role (interviewer or interviewee) and situation type (job interview or informal). Pearson's chi-square test, Fischer's exact test and Crosstabs test were performed to decide if there is a relationship between two categorial variables (e.g. between thematic role and pitch movement, position and pitch movement, discourse function and hand movement, etc.).

References

Abuczki, A. (2015). A Core/Periphery Approach to the Functional Spectrum of Discourse Markers in Multimodal Context. (Doktori értekezés.) Debreceni Egyetem.

Boersma P. & Weenink, D. (2007). *Praat: doing phonetics by computer 5.0.02.* University of Amsterdam: Institute of Phonetic Sciences. http://www.praat.org

Brugman, H. & Russel, A. (2004). Annotating multi-media / multi-modal resources with elan. In: Lino, M., Xavier, M., Ferreire, F., Costa, R., Silva, R. (Eds). *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)* (pp. 2065–2068). Lisbon: Portugal.

[2] SPSS Statistics is a popular software package used for statistical analysis; see at https://www.ibm.com/hu-en/marketplace/statistical-analysis-and-reporting.

'd Alessandro, C. & Mertens P. (2004). Prosogram: Semi-automatic Transcription of Prosody based on a Tonal Perception Model, In: B. Bel, I. Marlen (Eds.) *Proceedings of the 2nd International Conference of Speech Prosody.* Nara, 23 – 26 March 2004.

Fraser, B. (1990). An approach to discourse markers. *Journal of Pragmatics*, 14, 383–395.

Hunyadi, L., Földesi, A., Szekrényes, I., Staudt, A, Kiss, H., Abuczki, A. & Bódog A. (2012): Az ember-gép kommunikáció elméleti-technológiai modellje és nyelvtechnológiai vonatkozásai. In: Kenesei, I., Prószéky, G. & Várady T. (Eds.): *Általános Nyelvészeti Tanulmányok XXIV. Nyelvtechnológiai kutatások* (pp. 265–309). Budapest: Akadémiai Kiadó.

Hunyadi, L., Szekrényes I., Borbély A. & Kiss H. (2012). Annotation of spoken syntax in relation to prosody and multimodal pragmatics. In: *Proceedings of 3rd Cognitive Infocommunications Conference.* Kosice: IEEE Conference Publications. 537–541.

Mertens, P. (2004). Un outil pour la transcription de la prosodie dans les corpus oraoux. *Traitement Automatique des Langues*, 45 (2): 109–130.

Pápay, K., Szeghalmy Sz. & Szekrényes I. (2011). HuComTech Multimodal Corpus Annotation. *Argumentum*, 7. Debrecen: Debreceni Egyetemi Kiadó, 330–347.

Szekrényes, I., Csipkés L. & Oravecz Cs. (2011). A HuComTech-korpusz és -adatbázis számítógépes feldolgozási lehetbségei. Automatikus prozódiai annotáció. In: Tanács, A., Vincze V. (Eds.) *VIII. Magyar Számítógépes Nyelvészeti Konferencia* (pp. 190–198). Szeged: JATEPress.

Taylor, P. A. (1998). The tilt intonation model In: Proceedings of the International Conference on Spoken Language Processing, Sydney.

t' Hart, J. (1976) Psychoacoustic backgrounds of pitch contour stylisation, I.P.O. Annual Progress Report 11, 11–19.

Keyword extraction: its role in information processing

László Hunyadi

1) Preliminaries

WordNet, an important project as well as application of computational lexicography and lexicology assigns the following two senses to the term *key word*: a. a word that is used as a pattern to decode an encrypted message, b. a significant word used in indexing or cataloguing. The two senses are derived from its two components. Actually, we need a *key* to decode – and to understand, interpret – a message, and in order to do so we need to have some sort of a closer consideration of certain *words* which will be regarded significant in the given text.

This approach to interpretation suggests some of the most important issues associated with keyword extraction. When we are looking for a *key* in a message, we acknowledge that the surface, i.e. the sequence of words together with their associated default meaning cannot be compositionally "summed up" to amount to the sense our *understanding* of the given piece of text associates it with. It often happens that, due to the lack of background information or sufficient clarity of any kind we need to look for clues to identify the content, the sense, the interpretation behind the surface of the text. These clues can essentially be of two kinds: those which can be observed, described and studied appearing on the surface, and those which have a somewhat more abstract nature: the *relations* between these surface clues. All these clues are considered as keys to decode a message.

Although the term *key word* entails that it is words (and, as we saw above) their relations that are taken as clues to a text, the phrase *key word extraction* is often used in parallel with *content extraction* or *content analysis*. The alternative use of the term *content* points to the primary aim of keyword extraction: the decoding of a message essentially means that by doing so we wish to extract the relevant content of a text well beyond the words that we use as surface clues to it. Content, however, can be approached in several ways. A plausible way is *description*. This is what is usually required, among others, at school when a student is to relate a story, a play or a book. Such a description usually follows the form and structure of the text to be described consisting of properly organised sentences. However, often what is needed is to rather identify content with its subject matter which requires a somewhat substantial look at the text: what we would like to extract then is its essential cognitive classification. Of course, understanding and interpretation are cognitive processes, the ultimate aim of keyword extraction/content analysis can only be to provide the necessary and sufficient clues for this classification. It needs the identification and extraction of key words and phrases relevant and essential for the given subject matter together with the structures they occur in.

Accordingly, keyword extraction is an approach to the formal study of texts to derive their essential cognitive content hidden behind the surface. As such, keyword extraction is a field of *information retrieval* which, by its conception, shares approaches of information science, linguistics and textual studies in general.

2) Uses of keyword extraction

The development of keyword extraction as a field of information retrieval coincides with the rapid increase of textual information and the need to have control over it by efficiently understanding, classifying, selecting, using and further developing the huge amount of ideas presented in them. In the new electronic era we are witnessing ever newer forms and forums of presenting information while classical repositories are getting seriously overwhelmed with information, too. Our aim is then twofold: we need to preserve classical libraries and their classical collections by bringing to the surface and making available as much essential information as possible in a highly competitive situation, and we also need to follow the information flow of electronic channels in a similar fashion. Keyword extraction can play an essential role in pursuing these goals without distinction.

Without claiming to be exhaustive, we are going to mention a few significant areas where keyword extraction has already gained or is gaining an important role in information retrieval. We will group the numerous uses of keyword extraction around two main approaches: quantitative and qualitative which suggest the characteristic goals of the given uses:

i) Quantitative approaches

Automatic summarization

A classical form of content analysis which is employed in many areas of automatic text analysis. It essentially gives a selection of "important sentences" and generate a text as a summary from these sentences. It is based on keyword extraction and complemented by some kind of automatic text generation. Its uses include the creation of abstracts for further automatic processing. Collections of books, journal articles as well as submissions to electronic forums can effectively use automatic summarization.

Study of the internal cohesion of a text

In a world with an overwhelming amount of textual information it is highly recommended for a text to be as informative as possible. Its degree is mainly determined by the internal cohesion of the given text. In the case of a business report, e.g., the introduction should be highly informative, i.e. it should contain a high percentage of keywords and key phrases characteristic of the given field as well as the specificity of the given proposal. Keyword analysis is very useful in revealing the degree of cohesion and suggesting how effective the given text will convey the desired information.

Study of intertextual cohesion of texts

It is very important for texts with similar goals, such as follows-up of one and the same topic/correspondence to constitute to a single set of documents united by their subject matter and key reference. Cohesion of such similar topical texts can be checked as well as improved by this approach. It can effectively be combined with the study of the internal cohesion of the constituting texts.

Study of corpus homogeneity and similarity between corpora

It is an extension of the study of intertextual cohesion to a much larger number of texts with the aim to determine what set of texts will constitute a corpus. The determination of a corpus is essential in that a corpus often serves as an analysis basis for important decisions. The study of the similarity between corpora will help determine the proper extension of this corpus.

Authorship attribution

In literary and, in a broader sense, cultural studies it is one of the exciting issues to determine the author of a number of texts. Although, due to peculiarities of attributes of literature it is an issue of significant complexity, the study of the occurrence and distribution of keywords and phrases will contribute to the resolution of some of the related question.

Study of consistency of style or stance

Although authorship studies also include the study of the consistency of a given style, it may be misleading since one and the same author can choose to use more than one style across time. However, for a single work it is more probable to have a consistent style throughout the text. Accordingly, this use of keyword extraction can significantly contribute to stylistic studies in general. As a special case, forensic linguistics can also make efficient use of this approach in identifying textual manipulations, an important issue well beyond linguistic or literary studies.

The development of writing skills

Keyword analysis can be used to effectively contribute to the determination of the overall style of a text. As such, the study of the linear organisation of the text can help predict phrases to be used next in the text. This use of keyword analysis can contribute to developing writing skill as well as learning foreign languages.

Site content analysis

It is an emerging and very important area in which keyword extraction is expected to play a very significant role. Obviously, since an ever increasing amount of new pieces of textual information is published electronically which is specifically designed for the web or finds its way to the web, it is highly important to present or get hold of this information according to the needs of both the authors (owners) of the texts as well as those of the users. Accordingly, a site has to be characterized by its *relevance*. Relevance is an attribute of text typically of qualitative nature. However, this qualitative characteristic can be fairly reliably approached by studying the *density* parameters of certain keywords and by automatically *labeling* certain larger pieces of texts (usually web pages). Site content analysis has a huge economic importance since the classical economic relation of offer vs. demand is increasingly realised in the electronic relation of search vs. hit.

ii) Qualitative approaches

Content analysis

Although most of the uses of keyword extraction are aimed at finding some kind of statistical relation between observed surface data, there is a rapidly increasing need for an even more substantial investigation, when a kind of knowledge about the given field has also to be considered. In this sense we can talk about knowledge representation and, based on the representation of appropriate specific areas, knowledge extraction. Advances in this highly developing field involving several disciplines including logic, mathematics, information science, linguistics and more, demonstrate that however successful quantitative methods can prove to be, instead of giving some definitive answer to a content based question they only have a kind of approximation to them, and they are somewhat restrictive in nature being unable to allow for the relevance of certain questions emerging tasks may otherwise require. Obviously, automatic natural language processing (NLP) systems must be empowered with a well-defined set of knowledge about the given field in order to meet newer challenges. Accordingly, content analysis is a state-of-the-art field of NLP which has to be implemented in a framework where quantitative analysis is supplemented with the building of specific ontologies, descriptions of conceptual and knowledge based relations expressed by words, including keywords, and phrases with them within the given text. The inferences that are made on these grounds offer a broader description of what the given text is about than quantitative approaches alone. This approach (or such approaches) are already and will in future be even more increasingly helpful in the instantaneous content analysis and interpretation of electronic messages, tweets and blogs and – equipped with speech-to-text modules – real-time summaries of all sorts of conversations, speeches, and other verbal interactions.

3) Keyword extraction techniques

i) Quantitative techniques

Concordances

A concordance is the oldest tool for text analysis, dating back to the middle ages. Although a lot has changed since then and technical improvements – especially the introduction of computers – have resulted in a greatly improved performance to the extent that we can now formulate questions which, due to the technical limitations of earlier manual concordancing could not be asked at all, nevertheless the concordance as a research tool has not essentially changed in those centuries. It is, however, not a setback, it clearly shows that the concordance is a well-established and sufficiently proven tool in the hands of text analysers. Being still the most common tool for text analysis, it produces lists of words and phrases, it gives us the locations of the given word or phrase, and it offers elementary calculations on the basis of which words can be ranked and assigned the function of key words in its elementary sense.

Statistical relations and the organisation of the formal linguistic material

Although the primary ranking of words is done by some elementary statistics, the observation of more substantial statistical relations reveals significantly more about what lies behind the surface that we can directly observe. More advanced statistics can be applied to a set of wider range of data received from the linguistic parsing of words and phrases. *Morphological parsing* is used to identify lemmas of a text which enables us to describe the occurrence in the text of lexemes and roots rather than running words. Accordingly, the phrases "I work", "he works", "I am working", "He is working" will contain the occurrences of one and the same lexeme "work" with potentially one and the same meaning. The result of lemmatisation thus can be used to describe semantic and conceptual relations which is an important step towards the qualitative description of the given text. *Syntactic parsing* helps in PoS (part-of-speech) disambiguation which is crucial in determining the structure of a sentence. Morphological and syntactic parsing jointly help to determine the identification and the position of keywords in a text and, again, serve as an important input to semantic and conceptual analysis. Statistics can now be applied to these detailed data with much more strength: word prediction, idiom identification, anaphor resolution and much more is now supported by ranking, correlations, heuristics and other advanced statistical approaches.

ii) Qualitative techniques

Semantics

Since specific topics have specific words that are normally required for the given field, one might believe that a simple word list of a text will reveal what the text is actually about. However, although it is inevitable to study the occurrence of special words and expressions in a text, a predefined word list will, at best, only offer some very rough classification of the text. In addition, the study of such a predefined list of words will not allow us to subcategorize texts, understand their specific content matter in relation to similar ones, or identify and specify new areas of information. Accordingly, other, dynamic methods are also needed which take into consideration the actual organisation and development of a given text as it unfolds. Although, as we have seen above, the quantitative study of clearly formal parameters of text (including parsed linguistic patterns) is aimed at the qualitative description and analysis of a given text, the most crucial tasks of content analysis cannot be done without describing and analysing the *content* of the text. Although, paradoxically, we can use formal ("blind" in a way) methods to approach the content (the "message") of a text by making lists of words and rank them according to some predefined frequency gained from a large corpus, in this case content analysis *per se* is performed by our own reasoning. However, adding semantic and conceptual labelling to lexical items in the text and identifying the relations between them can result in a description of the text with formal semantic and conceptual structure. Statistics do not cease to be applied here, however, since, as we saw above, an enriched set of data enables us to apply a more sophisticated statistical formalism. This is what current research is really aimed at: to extract as much relevant information from the text as possible by using automation. The most significant development in the field of keyword extraction and content analysis can be witnessed in the application of the NLP-methodology to semantics as well as pragmatics. On the other hand, these new tasks also encourage theoretical linguists to propose and build semantic and conceptual frameworks which can give a feedback to and solve syntactic disambiguation as a prerequisite to finer formal textual analysis. Semantic analysis requires the creation of a semantics description of lexical items. As a result, the association of each lexical item with possible semantic domains together with their proper morphological and syntactic parsing assign certain relevant semantic/conceptual relations to phrases. Accordingly, the extraction of keywords will deliver highly structured semantic and conceptual relations which will refer to the content of a text with more detail, more specificity and more reliability than plain statistical methods resulting in an unqualified list of important words.

4) Where are we going

New tasks demand new efforts, very often new questions have to be asked to find the new ways to solve them. Whereas a concordance alone could solve many important questions related to a text in the somewhat distant past, our rapidly developing age appears to demand us to meet the challenge of real time content analysis. This, however, cannot be done overnight. We are now heading towards increasingly sophisticated and satisfactory models of keyword extraction, where the term keyword extraction is also gaining broader senses: we need to extract keywords in order to *identify* (in a way: understand) the content of many forms of textual communication, increasingly electronic and (still a bit further into the future) increasingly oral and increasingly online. Artificial intelligence is already increasingly employed along this path and we can anticipate it to occupy an even more central place in NLP. Previous achievements by linguistics will certainly be inherited, however AI technology and methodology may place new accents on them.

2

Teaching

Teaching by computer

István Csűry

This title might figure not at the top of a book chapter, and not even on the cover of a book but on the front pages of a book series. Indeed, we are facing a vast *interdisciplinary field* that no short overview could entirely cover in our current work. As the topic offers multiple entry points, we have chosen one of the most technical ones and will guide the reader toward the discovery of what is called today a virtual (or electronic) learning environment. To do so, let us take a closer look at *some software of relevant type* and introduce *basic technical terms*.

Evidently, the simplest general-purpose software can be integrated with classroom work or individual learning. A presentation tool or a software for interactive white boards (IWB) are more specific applications with respect to educational use. However, we will consider here neither the former nor the latter category. What we want to focus on are *tools intrinsically connected to teaching/learning contents as well as to the pedagogic process*, keeping in mind that they are a part of the larger framework of using information and communication technologies (ICT) for educational purposes. ICT in teaching/learning (as well as similar concepts like e-learning) has a rich and dynamically growing literature that you are suggested to discover autonomously, especially if you are interested in psychological, pedagogical, social and institutional issues it implies.

It should be stressed that solutions we are dealing with *exceed far beyond the institutional framework of schools*: e-learning is now everywhere, from schools and universities to self-training and from public life-long learning initiatives to human resources development at enterprises.

Another key point is that e-learning tools are usually designed *for users without computer programming skills*, i. e. for ordinary teachers or trainers, in such a way as to enable them to effectively produce teaching and/or testing material on user-friendly, intuitive interfaces.

From a technical point of view, we should distinguish (at least) the following components of a teaching/learning process:

- structuring and presentation of knowledge
- learners' work (individually or in interaction with teachers and/or other learners), including learning and evaluative exercises
- organizing the educational/training process, including learner administration and tracking learners' scores

Educational software begins where the second component, i. e. learners' work is taken into account. In addition to task development tools, more than one combination does exist: course authoring tools combined with exercise-making functions or full-fledged course management systems provided with everything one might need during a whole teaching/learning cycle.

1) Course Authoring and Exercise/Test Development

i) Making automated learning and evaluative exercises

Automating exercises and tests: pros, cons and hows

In a computerized framework, teachers or course creators write exercises with (all) good and bad answers incorporated and with points and feedback assigned to each. In addition, they may include illustrations, resources, navigation possibilities and timing.

There are several evident advantages of creating and using computerized, self-correcting exercises:

- The possibility of using of any kind of *media* and of various (online) *resources* lifts off the limitations of traditional course material and is a factor of enrichment, dynamism and motivation.
- Learners benefit from *instant correction* of their answers (especially if the teacher provides *explanatory feedback* for wrong answers).
- It extends teachers' capacity to *handle individually learners' needs* and enables the latter to work at their own pace.
- *Mechanical tasks* of correcting, explaining and evaluating are reduced, leaving more time for creative work.
- The *layout* of the questions or exercises is easy to define inasmuch as the tool automatically sets structures inherent to specific activity types. (For example, you have not to manually format the answers of a multiple-choice question or align checkboxes with them.) What is more, you can even export a quiz for printing, which makes an e-learning editor a better choice for this type of task than a word processor.
- You do not have to care about *randomizing*: the system may display options (as answers for a multiple-choice question, for example) in a different order every time the question is viewed. It is also possible to give students different questions randomly selected from a set of questions.
- Once you have created some question or activity, you can reuse it as needed, even by *exporting* it into different formats for *importing* in another software or system. Course authoring and management systems rely on standardized formats and *norms* (like xml, SCORM or AICC) which allow easy transfer of contents.

Some disadvantages could also be noted as, for instance, the somewhat rigid character of multiple-choice exercises or the difficulty of foreseeing every possible answer to an open question. However, in integrated course management systems, *open activities* (like essays) are allowed and learners and teachers may interact directly on such tasks, which is still a comfortable solution for non-uniformizable knowledge. Besides, e-learning tools are good for help to realize ideas and not for relieve anybody of thinking: one should use them advisedly in the framework of a well-thought-through pedagogical project.

Usual types of questions/activities

a) *True/False* question[1]: learners are given only these two choices for an answer. Remember that this question type may be seen as a variant of a multiple-choice question. When editing a question of this type, we need to formulate the question, a good answer and a wrong one, and to mark for the system which is the good answer.

b) *Multiple-choice* question (or *single-answer* question): learners may choose one and only one answer by clicking on radio buttons or ticking in checkboxes next to the answers. When editing a question of this type, we need to formulate the question, a good answer and several wrong ones, and to mark for the system which is the good answer.

c) *Multiple-answer* question: learners may choose more than one answer by clicking on radio buttons or ticking in checkboxes next to the answers. When editing a question of this type, we need to formulate the question, the good answers and one wrong answer or more, and to mark for the system which are the good answers. Remember that the same question editing tool may allow the creation of the two latter types by adequate setting possibilities.

d) *Matching* question: learners are presented with two lists of content. The items (names, statements, etc.) of one of the lists must be correctly matched against those of the other one. For example, learners may have to align dates with historical events or authors with the titles of their works. Matching items may be presented on drag-and-drop cards or in a form of dropdown menus placed next to the items of the first column. From the point of view of editing, that does not make much difference as in each case, the teacher has to enter the matching parts in the corresponding cases of a blank form. – A special version of the matching question type is a drag-and-drop onto image activity where learners have to place labels on images, e. g. for identifying the parts of a structure.

e) *Short answer* question: this type is a semi-open question as learners have to enter the answer themselves by typing in a word or phrase in a text box. As the character string they submit must exactly match one of the acceptable answers previously listed by the teacher or course creator, the

[1] The term *question* is used in an abstract sense: in fact, sentences or text learners are presented with may take other than interrogative forms.

use of this question type is limited to the cases where all possible (good, but maybe also bad) answers are foreseeable. Editors may be allowed to set a slight level of tolerance or even some variables in order to deal more efficiently with misspelling, variants and letter case.

f) *Cloze test* or *gap-filling* exercise: the place of some words or expressions in a phrase or a more or less longer text is left blank; learners have to fill in these gaps. There are several techniques: the list of the omitted words may be or not presented to the learner. If this word list is given, it may appear either somewhere on the page as an unordered set of words or as a drop-down list in every gap to fill. As gapped text usually may have rich text features (e. g. tabular arrangement, graphics, etc), this is a versatile activity type where every editing option has pedagogical consequences. In some systems, gaps are created by using a simple graphical interface whereas in other cases, some minimal coding may be necessary, i. e. the gaps and options are to be edited as mere character strings.

g) *Reordering* question: elements of the answer are given in disorder and learners must put them in the correct order by successively clicking on them or by dragging and dropping them. This type is also known as a jumbled sentence exercise; nevertheless, it can be used not only as a grammar exercise in a language course.

h) *Crossword*: this type of exercise is simply the computerized version of traditional crossword puzzles. While editing, teachers have only to enter the words and the clues or definitions in a blank form and may let the system arrange both across and down the numbered squares to be filled with the letters of the given words. As simple to create as it is, this kind of exercise yields a gamified yet very useful way to control and develop vocabulary or terminology knowledge.

There are other types of questions/exercises or variants of the enumerated ones. Some of them, used for example for teaching/learning mathematics, require numbers as answers. In *calculated* questions, we can use common variables names as *x*, *y* that the system substitutes with random values whenever a learner takes the quiz. The correct answer is calculated by the system according to a formula given by the editor of the question.

Others are *open questions* for which no unique correct answer might be included. Teachers formulate the question and/or some indications, learners having to enter their answer in a text box. Answers may span from a sentence or a paragraph to an entire essay. Sometimes students may/must upload their answer as a separate file. It should be noted that creating open questions is pointless elsewhere than in course management systems offering the possibility of a "live" feedback.

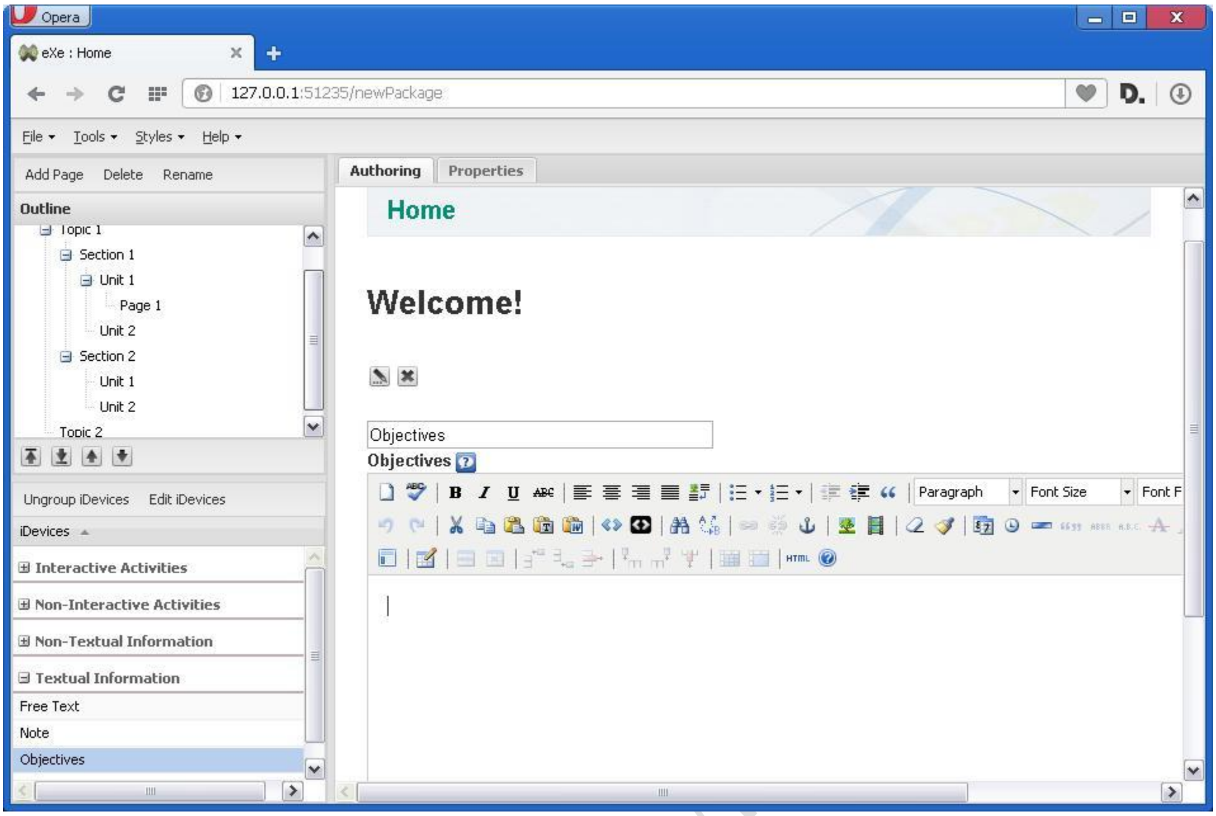
Although we are interested in technical and not didactical aspects of e-learning, we would like to emphasize the *functional rather than mechanical use* of the different question types. Even more, feedback, weighing of points or grades for good and wrong answers or the way wrong answers are conceived all affect the success of the teaching/learning process.

ii) Editors for course and/or activity authoring

There is enough similarity between different systems in basic editing functions for anyone having acquired some skills with one of them may confidently try out another while searching for better solutions in terms of fine-tuning activities and interactions, enhancing interoperability with other (e. g. presentation) software and use of multimedia, or user interface. The following examples illustrate these basic functionalities.

First of all, one would present content to be studied in various, well-structured forms. Content may be organized as a book (with a hierarchical structure) or as a series of units or pages, or maybe as a combination of the two. There may be different content types: text, graphics, images, sounds, video, external resources, etc. A course editor has to handle all these.

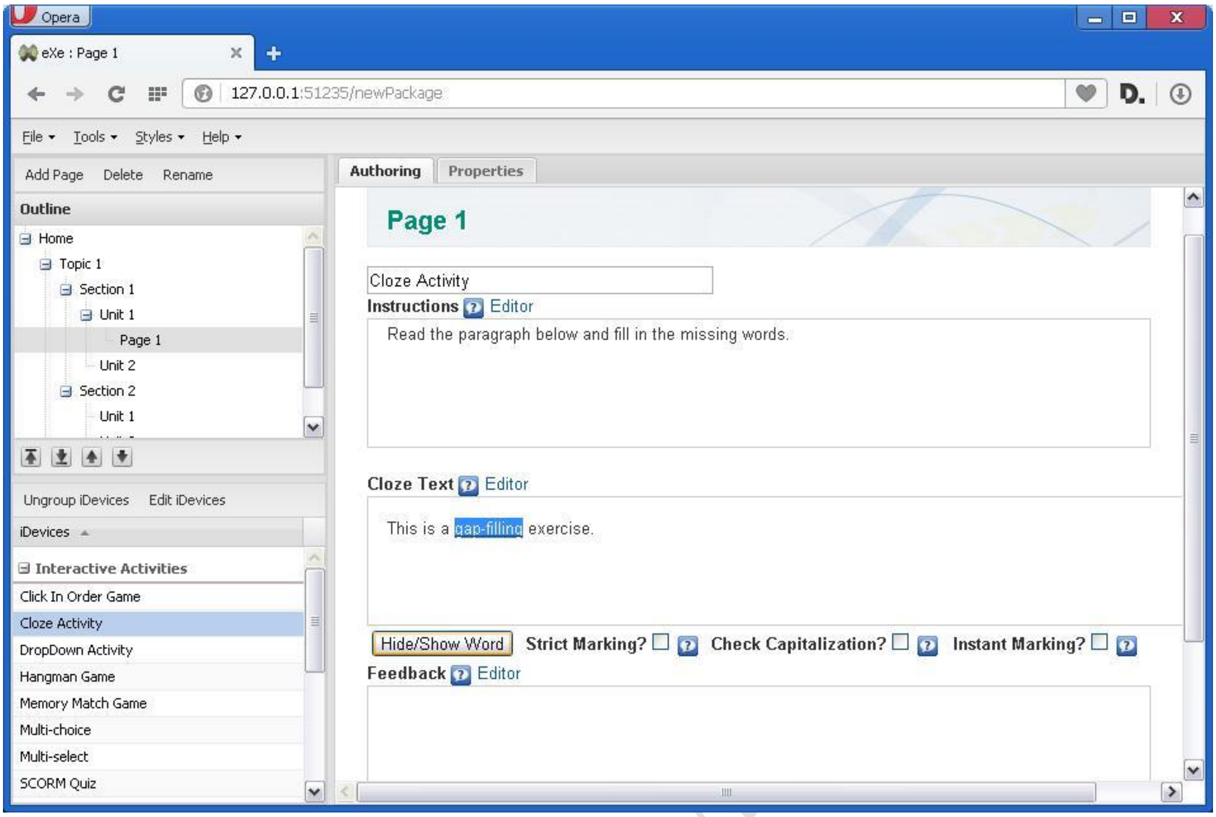
The following screenshot shows the graphical interface of a software called eXe Editor. There is a course package being edited that has a structure of two topics, with two sections in the first one and two units in each section. This hierarchical structure, i. e. the outline of the course package appears in the upper part of the left pane where are several buttons at our disposal for collapsing or expanding the different levels of the structure as well as for modifying the place of its items, adding new items or deleting existing ones, or simply renaming them. The larger part of the screen shows the content of the highlighted structure item as a webpage. In this case, we can see the "Home" page, at the root of the structure, to which a "Welcome" message has already been added, and that will present the objectives of the course package to the learners. The "Objectives" module, which is just being edited in a rich text editor window, has been added by clicking on the corresponding button of the lower part of the left pane. This region of the graphical interface enables the user to add textual as well as non-textual information and interactive as well as non-interactive activities. Commands in the right pane allow for editing, rearranging or deleting content belonging to a given structural level.



Screenshot 1

The next screenshot shows how a Cloze activity is chosen among the interactive activities and added to the Page 1 of the Unit 1 of Topic 1, Section 1. We can edit instructions and feedback in separate textboxes, highlight words of the text of the exercise and mark them as to be gapped with a single click on a button, and set the behaviour of the exercise in some respects. We can get hints on using commands by clicking on the question mark icons next to the buttons and call the rich text editor at any time if we would like to enhance the look of the exercise, add illustrations to it, etc.

The third screenshot presents the result when the package has been exported as a folder of html pages. As we can see, a successful attempt has been made to solve the exercise. Automatically generated navigation links correspond to the structural items defined in the editor.

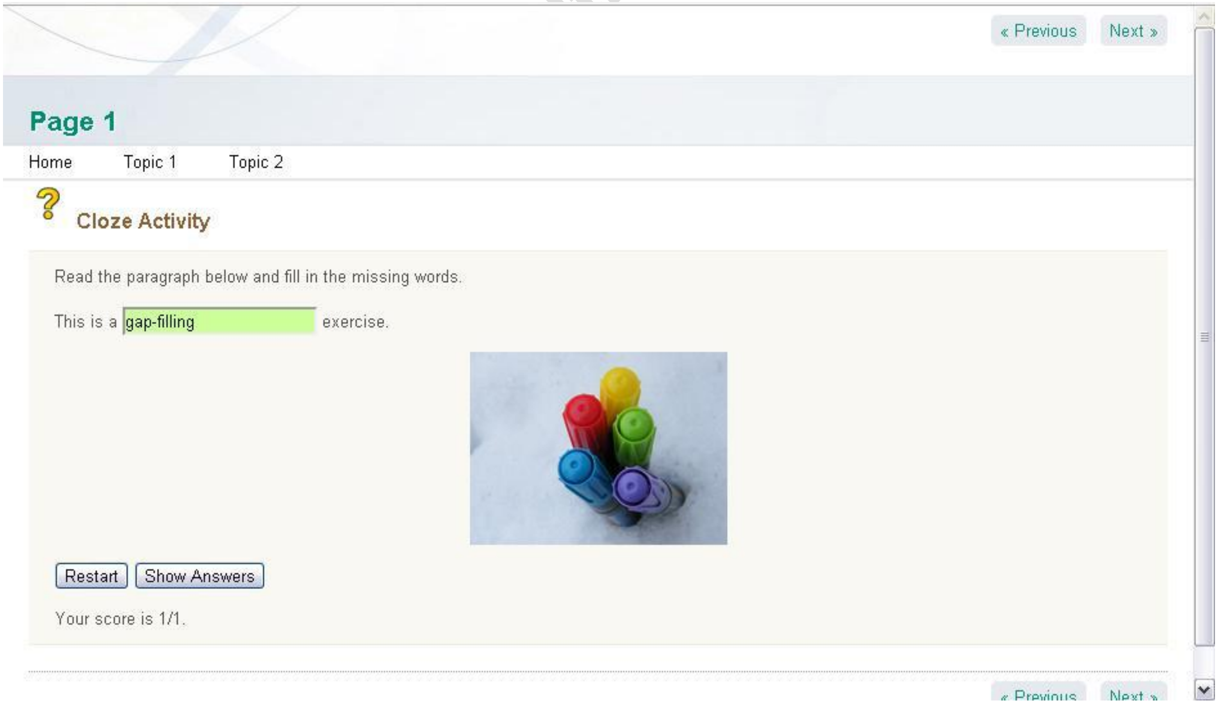


Cloze Activity

Read the paragraph below and fill in the missing words.

This is a gap-filling exercise.

Screenshot 2



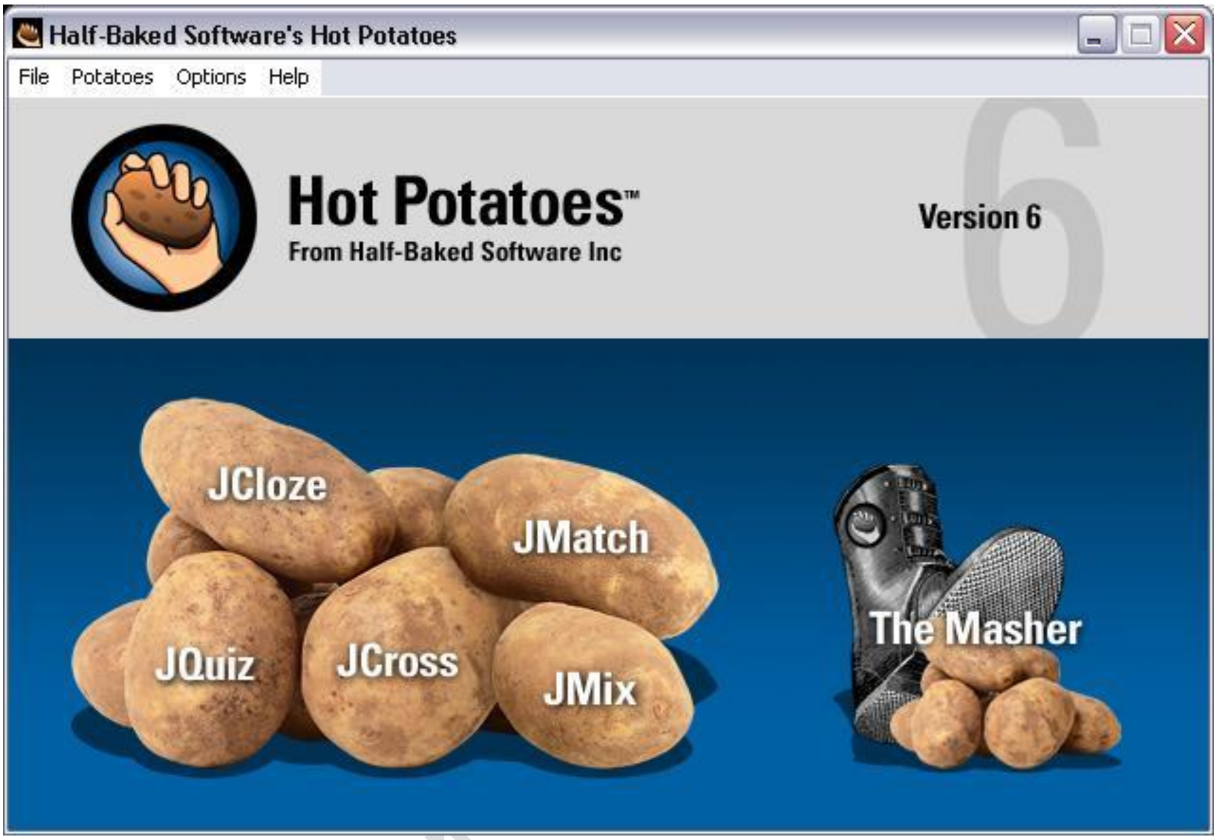
Read the paragraph below and fill in the missing words.

This is a gap-filling exercise.

Your score is 1/1.

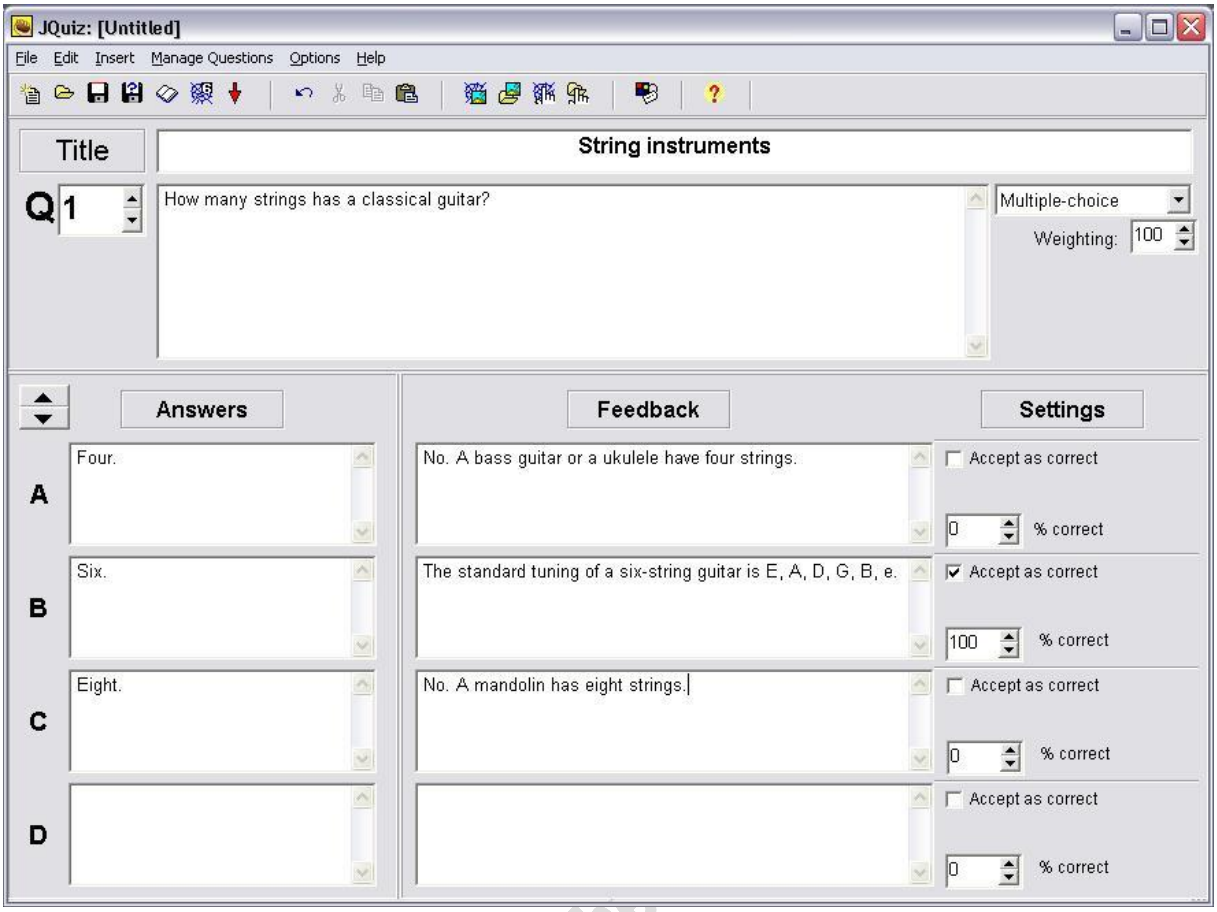
Screenshot 3

Compared to this *top-to-bottom* approach, you may want to proceed in the adverse way by starting to edit exercises prior to putting them into some teaching material, online course, etc. Other software, like Hot Potatoes, follow the latter, *bottom-to-top* approach. As it is shown on screenshot 4, this software contains five tools for five (or more) different types of exercises (gap filling, matching exercises, short-answer or multiple choice/multiple answer quizzes, crossword puzzles and jumbled sentence exercises) as well as a sixth one allowing the composition of suites of activities.



Screenshot 4

The figure below shows the quiz-editing interface of Hot Potatoes. A quiz on string instruments is being edited; it is the first question that we see. As one might guess looking at the counter at the top on the left, the tool enables the creation of an exercise consisting of a whole set of questions, unlike some other systems where we have to put quiz questions together (in a test, for example) after having created them one by one. The number of answers may be freely set and vary from one question to another, and every question of a set may be weighted differently in order to reflect its importance in a given unit of knowledge. Question type (multiple-choice, multiple-select, short-answer or hybrid) is chosen by using a drop-down menu.



Title: String instruments

Q 1: How many strings has a classical guitar?

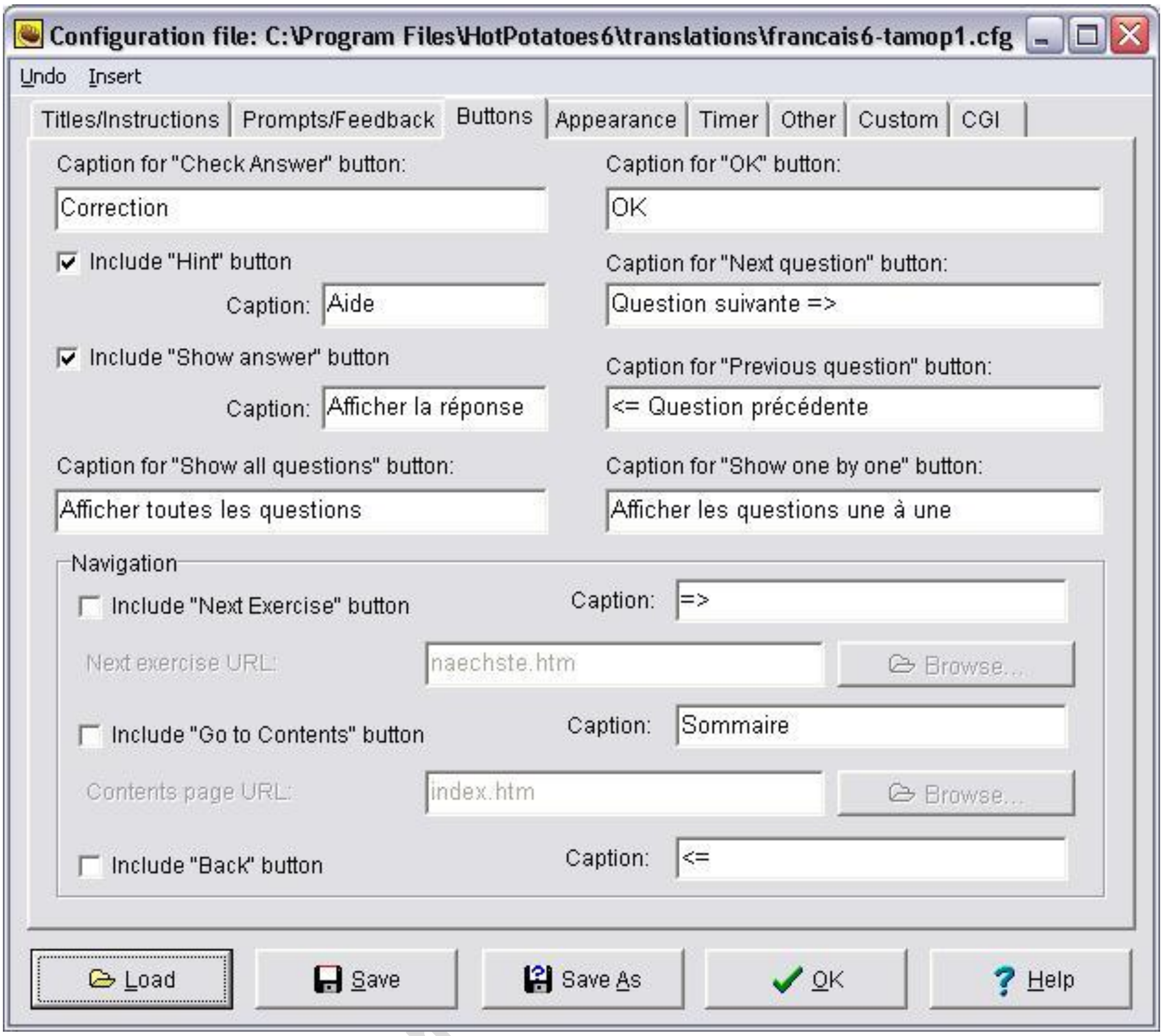
Multiple-choice

Weighting: 100

	Answers	Feedback	Settings
A	Four.	No. A bass guitar or a ukulele have four strings.	□ Accept as correct 0 % correct
B	Six.	The standard tuning of a six-string guitar is E, A, D, G, B, e.	☑ Accept as correct 100 % correct
C	Eight.	No. A mandolin has eight strings.	□ Accept as correct 0 % correct
D			□ Accept as correct 0 % correct

Screenshot 5

Many options are at our disposal in the main menu for an enhanced question editing and a flexible management of the set of questions in the exercise. We may insert tables, images or other media as well as web links in the question field, and there is a rich bunch of settings that allow a fine-tuned configuration of the output. Screenshot 6 shows the dialog box with controls for setting exercise titles, instructions, general feedback and other contextual messages, buttons and commands to be inserted in the exercise page as well as its look, a timer if needed, and much more. Configurations of all these settings may be saved in files one can simply reload in case of similar editing tasks.



Caption for "Check Answer" button: Correction

☑ Include "Hint" button
Caption: Aide

☑ Include "Show answer" button
Caption: Afficher la réponse

Caption for "Show all questions" button: Afficher toutes les questions

Caption for "OK" button: OK

Caption for "Next question" button: Question suivante =>

Caption for "Previous question" button: <= Question précédente

Caption for "Show one by one" button: Afficher les questions une à une

Navigation

☐ Include "Next Exercise" button — Caption: =>
Next exercise URL: naechste.htm — Browse...

☐ Include "Go to Contents" button — Caption: Sommaire
Contents page URL: index.htm — Browse...

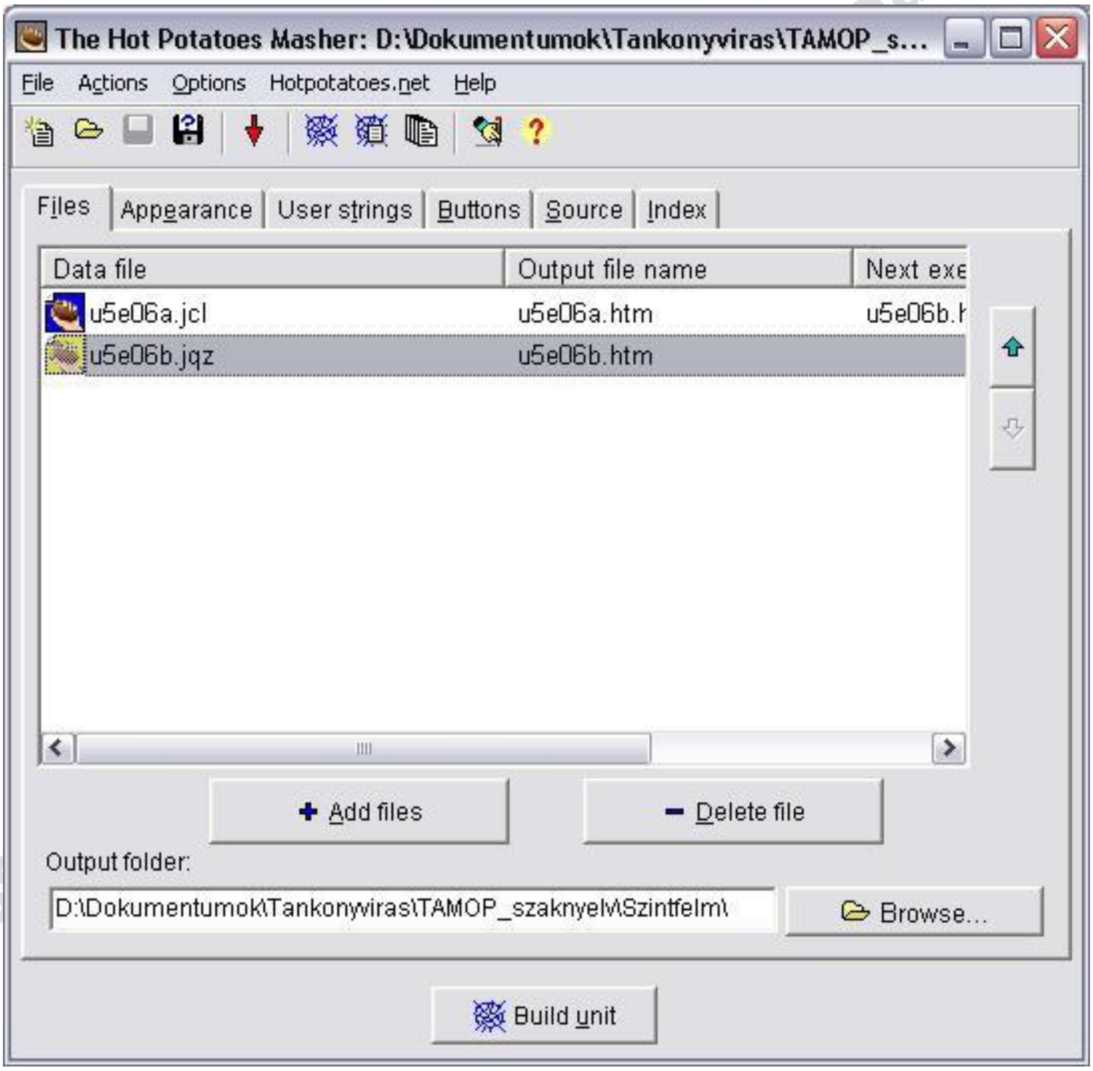
☐ Include "Back" button — Caption: <=

Screenshot 6

The author of the exercises has the possibility of exporting the result of his work in various formats. (As one may have guessed, exporting is a necessary step here given that course or exercise editors are only for authors and not for learners: the latter will do the exercises on some independent platform, typically in a web browser.). In Hot Potatoes, the simplest solution is to export for printing. By doing this way, we copy our exercise to the clipboard; further formatting and printing can be carried out after pasting it into a word processor. Obviously, we need to keep the exercise in a digital format if we want it to be interactive. It may be done either by exporting it as a web page (interactive functions are then coded in Java scripts) or by creating a SCORM package that one wants to upload in a course management system.

The "Masher" tool in Hot Potatoes enables users to put their various exercises made with the other tools together, in a structured form, with navigation possibilities, that they can save as a project for further editing with the software or export to html or SCORM. As we can see on Screenshot 7, editing the suite of exercises is done through an intuitive interface.

As we have seen, besides the possibility of creating roughly identical questions or exercises, the two software presented above display notable differences even beyond the contrast between the top-to-bottom and the bottom-to-top approach. Hot Potatoes has outstanding features for creating a good variety of fine-tuned interactive exercises in an efficient way but has somewhat more limited capabilities of editing a course package, especially as for its non-interactive part. On the other hand, eXe Editor seems less powerful in question/exercise editing but it is a better tool for building a more complex e-learning material. As a conclusion, *it is always the teacher's (or course creator's) task to choose an e-learning editing tool according to his actual goals*.



Data file	Output file name	Next exe
u5e06a.jcl	u5e06a.htm	u5e06b.h
u5e06b.jqz	u5e06b.htm	

Output folder:

D:\Dokumentumok\Tankonyviras\TAMOP_szaknyelv\Szintfelm\

Screenshot 7

iii) Some examples of course (or exercise) authoring software

Free/open source software

name of software	author / publisher / company; website	main features
Hot Potatoes	Half-Baked Software Inc.; https://hotpot.uvic.ca/	(as presented above; no longer developed/supported but still available)
QuizFaber	Luca Galli; http://www.lucagalli.net/quizfaber	• creates multimedia quizzes as HTML documents • types of questions: multiple-choice, multiple-answers, true or false, open questions, gap filling exercises, matching words • embedding multimedia objects • printing and exporting in many formats (among which pdf)
eXe	University of Auckland, The Auckland University of Technology, and Tairawhiti Polytechnic; http://exelearning.org/	• offline authoring environment • enables teachers to publish structures of professional looking web pages with multimedia content for learning • i-devices (instructional d.) for adding interactive and non-interactive tasks • types of questions: multiple-choice, multiple-answers, true or false, open questions, gap filling exercises, sort items, hangman game, memory match game • printing; exporting as text or as web site as well as in educational standard formats, Ustad Mobile, Epub and XLIFF
Xerte Project	Xerte, an Apereo community project; http://www.xerte.org.uk/	• full suite of tools • Xerte: fully-featured e-learning development environment for developers creating sophisticated content with rich interactivity • Xerte Online Toolkits: server-based suite of tools for content authors producing interactive learning materials

Commercial software

name of software	author / publisher / company; website	main features
Articulate Storyline 2	Articulate Global, Inc.; https://www.articulate.com/products/storyline-why.php	• creates interactions in personalized ways • creates 25 different question types • publishes content for nearly any device • exports to files according to e-learning standards • market-leading software used by top organizations • free trial
Udutu Authoring Tool	Udutu Learning Systems Inc.; http://www.udutu.com/	• cloud based authoring software • supports any web compatible content • pre-built game-like scenario templates • interactive assessment templates • exports to html and SCORM • several formulas from free to full-featured • the company sells an LMS too
CourseLab	WebSoft Ltd.; http://www.courselab.com/	• creates all kind of interactive e-learning content and a wide range of applications (e.g. assessments, quizzes, surveys, etc) • supports any Internet-friendly data formats • contains a large library of ready-to-use complex objects • PowerPoint-like user interface • produces the most current question types • exports to files according to e-learning standards • distributed in a free version as well, with limited features

6) Course (or Learning) Management Systems (CMS/LMS) or virtual learning environments (VLE)

i) What kind of software CMSs/LMSs/VLEs are?

Although they denote somewhat different visions, the three terms in the section title are synonymous and refer to the same kind of software system.[2] They belong

[2] As not all systems offer the same functionalities, however, there may be some discrepancy in different uses of these terms. For example, one might call LMS only software that is not used to

to terminological traditions of different countries or geographical/cultural areas. By using the abbreviation CMS in the following for the sake of simplicity, we do not intend to express any preference. Our choice, however, may stress the relatedness of such e-learning systems with general-purpose *c*ontent *m*anagement *s*ystems. Indeed, a learning management system can be seen as a specialized type of content management systems. To put it simply, a CMS is a *software installed on a server and accessed via Internet by the different actors of training/learning processes*, such as course managers, course or content creators, teachers and students in a given organizational setting. In other words, CMSs, unlike software we have seen above, is not a tool for individual work on a teacher's workstation or personal laptop. This software is *the framework of participants' interactions and is intended to serve them by several ways in technically handling their teaching/learning activities*.

ii) Main characteristics and functionalities of VLEs/CMSs/LMSs

A CMS is usually implemented within *educational institutions* or in a *corporate setting* (where it is used as a means of human resources development, for personnel training programs). It provides a *web-based platform for all aspects of study courses*: course and content editing and delivery, administration/documentation of learner enrolments and activity, online collaboration possibilities between all participants of the teaching process, assessments and tracking of results, analyse and reporting on various aspects of programs (for example, skills gap analysis or evaluation of the efficiency of questions in tests). Evidently, not every CMSs is equipped with every possible feature, and modular ones may be implemented with different sets of functionalities from one installation to another.

In order to get access to the system, users have to register and are given *roles* according to their tasks and authorizations. Normally, users can enter the system by using their personal ID and password. *Management of users and access rights* is one of the essential features of a CMS. A system administrator is at the top of the hierarchy; users with this role are responsible for running the system in the technical sense of the term. Typical roles are managers, course creators, teachers, students and guests. Students may be set into groups for group activities, and cohorts may be created according to training cycles (e.g. for regrouping students of the same course in different years).

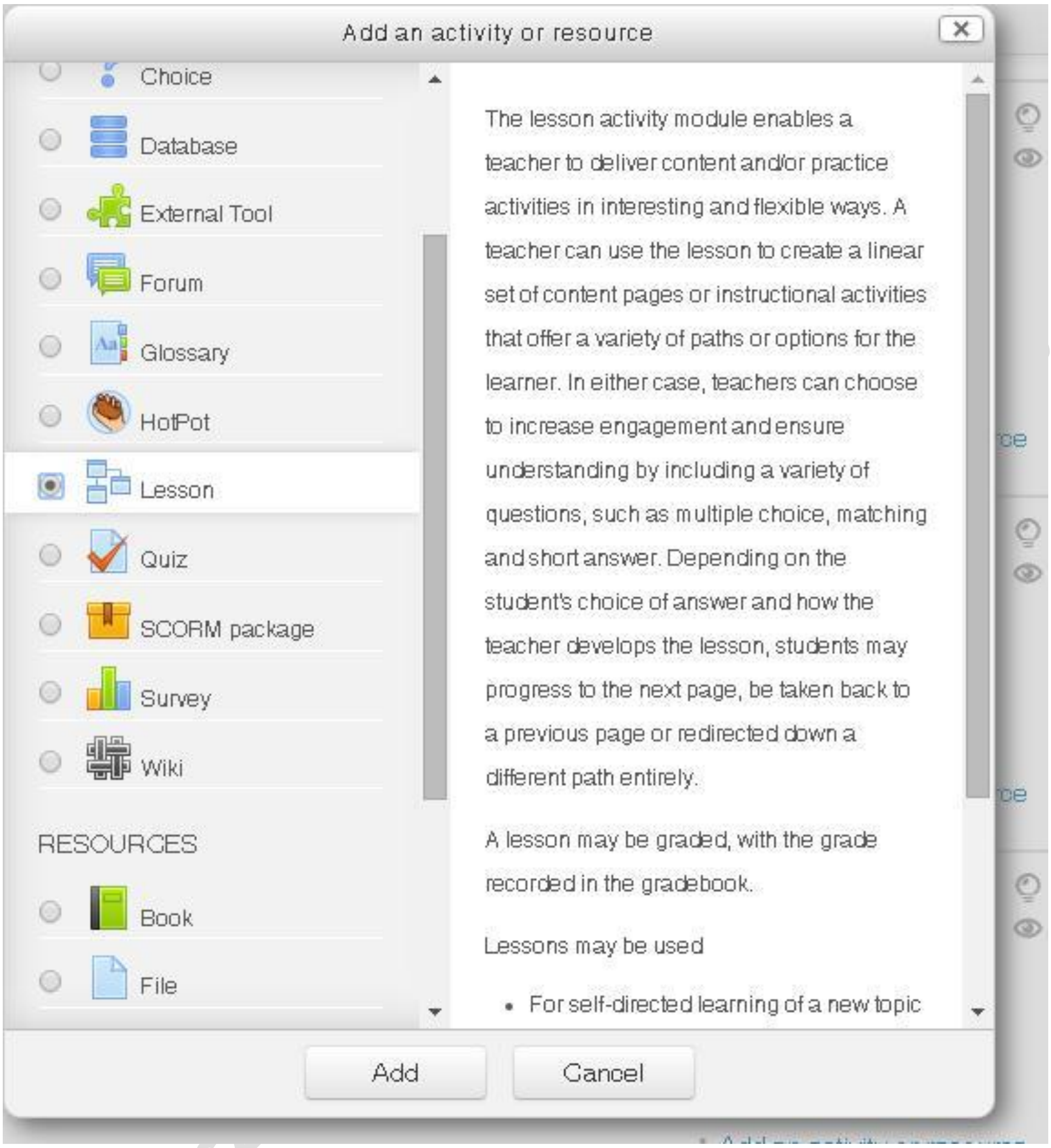
Security of data being an essential issue, CMSs usually rely on secured communication protocols. The system may be integrated to some extent with other systems and databases in use at the given institution (e.g. the system of student administration in a university).

create course content. Nevertheless, integration being at some point inevitable, we prefer not to make such subtleties the object of a discussion in extenso.

As for the core activities, such systems of integrated teaching/learning and course management tools typically feature the following functionalities:

- creating and managing *courses*, contingently grouped into course categories (N.B. a course is a unit having well-defined learning/teaching goals, a structure, a schedule, contents, learning/teaching activities, assessments and participants)
- adding and managing *contents* as learning materials either by linking external sources, embedding documents edited with other software or by creating them with the system's built-in editing facilities
- creating and managing *course activities*, i. e. various learning and evaluative tasks to be fulfilled by students in interaction with the system (and maybe with the other participants)
- developing and managing a *question bank* consisting of various types of questions/exercises, usually organized in categories, that may be used in learning as well as evaluative activities
- defining time (and other) *constraints* that must be observed by students in accomplishing their tasks
- defining *learning itineraries* by setting the fulfillment with a required minimum score of certain tasks as prerequisites to access to further learning units and by orienting students to learning activities according to their detected skills and skill gaps
- *evaluation and grading* of students' work all along their activities as well as *tracking* of their results
- *statistical analysis* of students' activities and their results obtained at specific questions allowing for improvements to the teaching and testing materials
- *collaboration* and *communication* among participants via messaging and forums
- organizing *teamwork* of teachers/tutors of the same course

Screenshot 8 shows a dialog box for inserting content (resources) and activities in the Moodle system.



The lesson activity module enables a teacher to deliver content and/or practice activities in interesting and flexible ways. A teacher can use the lesson to create a linear set of content pages or instructional activities that offer a variety of paths or options for the learner. In either case, teachers can choose to increase engagement and ensure understanding by including a variety of questions, such as multiple choice, matching and short answer. Depending on the student's choice of answer and how the teacher develops the lesson, students may progress to the next page, be taken back to a previous page or redirected down a different path entirely.

A lesson may be graded, with the grade recorded in the gradebook.

Lessons may be used

- For self-directed learning of a new topic

Screenshot 8

iii) Examples of CMS/LMS/VLE software

The number of CMS/LMS/VLE solutions offered by different companies, institutions and communities over the Internet indicates what an important industry e-learning is. Giving an overview of the market would be far beyond the limits of this chapter, let us therefore content ourselves with a brief comparison of two systems among the most widely used ones. (N.B.: choosing and installing such a system usually depends on an organizational decision.)

Free/open source software

name of software	author / publisher / company; website	main features
Moodle	Moodle Pty Ltd; https://moodle.org/	• the world's most popular LMS • robust open-source learning platform supported by a global community – but you have to support it on your own (system administration) • collaborative tools and activities • intuitive interface easy to navigate on both desktop and mobile devices with a personalized dashboard, all-in-one calendar and notifications • supports open standards; modular structure with high interoperability; customizable site design and layout; multilingual capability

Commercial software

name of software	author / publisher / company; website	main features
Blackboard	Blackboard Inc.; http://www.blackboard.com	• one of the market leader proprietary systems • functionalities divided among several software that you can buy following your needs (which may be not only educative): Blackboard Learn (the VLE itself), Blackboard Collaborate (real-time online collaboration platform), Blackboard Connect (two-way messaging with the members of a Bb community), Blackboard Mobile (extends the VLE to mobile platforms) and Blackboard Analytics (facilitates decision-making by bringing data together) • customer support (price can also include the cost of hosting or server management together with the license fees)

Further reading

Clark, Ruth C. and Mayer, Richard E. (2011^3) *E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning.* Pfeiffer essential resources for training and HR professionals. San Francisco: John Wiley & Sons

Garrison, D. Randy (2011) *E-Learning in the 21st Century: A Framework for Research and Practice*. New York: Routledge

Haythornthwaite, Caroline and Andrews, Richard (2011) *E-learning Theory and Practice*. Thousand Oaks: SAGE Publications

Khan, Badrul Huda (ed.) (2005) *Managing E-learning: Design, Delivery, Implementation, and Evaluation*. Hershey, London: Information Science Publishing

3

Translation

Computers in Translation

István Csűry

Translation is a main issue in several fields of the humanities, and so it is for computing. Research topics as well as computerized solutions and applications cover such a large scale that it would hardly be possible to present here a comprehensive overview. As widespread as it might be, however, even in everyday use of the Internet (see, for example, Google Translator), informatics in translation is widely ignored and subject to general misconception and misunderstanding. Hence, the aim of this chapter is to provide readers with a first overall orientation and with simple definitions of the most elementary notions.

1) Two (?) directions in using computer for translation

i) Machine (or Automatic) Translation

Properly used, the term *machine translation* (MT) refers to translation processes of written discourse entirely carried out by computer software, with no involvement of human translators at all. Synonymous terms are automated / automatic or instant translation. Research in this field has evolved in parallel with computers and informatics. As we will show below, MT is a far more serious business than average Internet users might judge it after having been confronted with some funny results of automated translation.

ii) Cases when CAT is not an animal

Evidently, there is no (professional) translation today without computers. Nevertheless, *C*omputer *A*ided (or *A*ssisted) *T*ranslation refers to specific solutions applied systematically to translation-related tasks and could not be reduced to haphazard use of software and networks during translation. Three (closely linked) aspects of translation may be concerned:

- language-oriented operations
- management of the translation process as a service
- integration of software tools

The first aspect implies translation memory usage, search in dictionaries and terminology lookup as well as different kinds of editing processes. The second involves communication and administrative work, while the third concerns technical aspects of creating and maintaining a computerized environment of well collaborating software for the whole workflow. (See for some detail and illustration below, in section iii), page 126.)

iii) And why the question mark? Convergence between CAT and MT

Despite of the definitions given above, the limit between MT and CAT is not as sharp as it seems at the first glance. Combining CAT with machine translation was a logical step from both sides, i.e. from the point of view of MT as well as in a CAT approach. When used in a professional setting, the first (MT) is normally completed by revising/post-editing done by human translators. In other words, while the MT software runs effectively without human involvement, its output is considered as a draft and translators intervene however in the translation process as a whole. As for the second (CAT), the presence of an auto-translate function for 100% matching segments in translation memory software shows that MT is conceived as a potential part of CAT solutions and it may occur in various stages of the workflow. Finally, we should note that translation memory databases (or translation memory-like data/corpora) enter in the constitution of statistical MT systems (see below for explanation). Therefore, regardless of its autonomous uses, MT is rather a (possible) part of CAT in the universe of professional translation.

2) Translation memory software

i) General notions

What does a translation memory look like?

A translation memory (TM) is a language-independent tool the use of which became not only generalized but also mandatory in translation industry. The term itself has unfortunately two senses. In the first, broader one, it refers to a specific type of software offering a framework for translating operations. In the second, more specific sense, a TM is a database (storable and exchangeable in the form of a computer file) that is at the heart of TM software. In fact, a TM utilizes existing translations in order to spare its user translating anew already translated segments of texts. These segments are organized in a database that continually grows by capturing for future use every new translation that its users produce. As database files, TMs have specific structure and markup designed with respect to interchangeability and standards as TMX (Translation Memory eXchange – an XML specification allowing the exchange of TM data).

A TM database stores translation units (TUs). A TU consists of three kinds of data:

- a segment of text (typically a sentence, but also titles, captions, etc) in source language,
- its translation in target language and
- some metadata, i.e. information about the pair of segments, like identification of the source and the target languages.

There are two ways of creating a TM: either by aligning the segments of an existing source-and-translation pair of texts or by starting to build a TM from scratch with an "empty" TM software: the first TU will then be created when you have translated the first segment of your first source text. Any further use of a TM database builds it up more and more, increasing productivity this way.

As software, TMs are usually integrated with other tools, such as a word processor, an electronic bi- or multilingual dictionary or a terminology management system. As we have seen, it may even be linked with a MT system.

Possible uses and benefits of using a TM

Quite obviously, as repetitiveness is a key factor in TM usage, this kind of tool is not suitable for literary translation, which has not to deal with a significant amount of repeated segments of text. (N.B.: TUs contain rather complex meaningful units and not just words or syntagms.) Technical discourse is, on the contrary, full of repetitions. Repeated segments may also occur inside the same text, but even more among different ones belonging to the same text type. For example, contracts, product descriptions, user manuals, annual reports or minutes of general assembly meetings of companies are very similar from a linguistic point of view. That is the kind of discourse a professional translator has to cope with, who may expect the following advantages from using TMs:

- accelerated workflow and increased productivity thanks to (semi-) automatic translation of already translated segments of text
- better quality assurance thanks to a consistent usage of terminology and phraseology
- possibility of efficient teamwork with other translators on larger or similar projects (and, consequently, use of very large TMs)
- benefits of an ergonomic interface (everything you need is "at hand" and neatly displayed on your screen)
- less effort spent on "surface" activities like the typographical layout of texts.

ii) Main functionalities of TMs

Suppose that you have a convenient TM for your actual translation job. The translation process will run as follows:

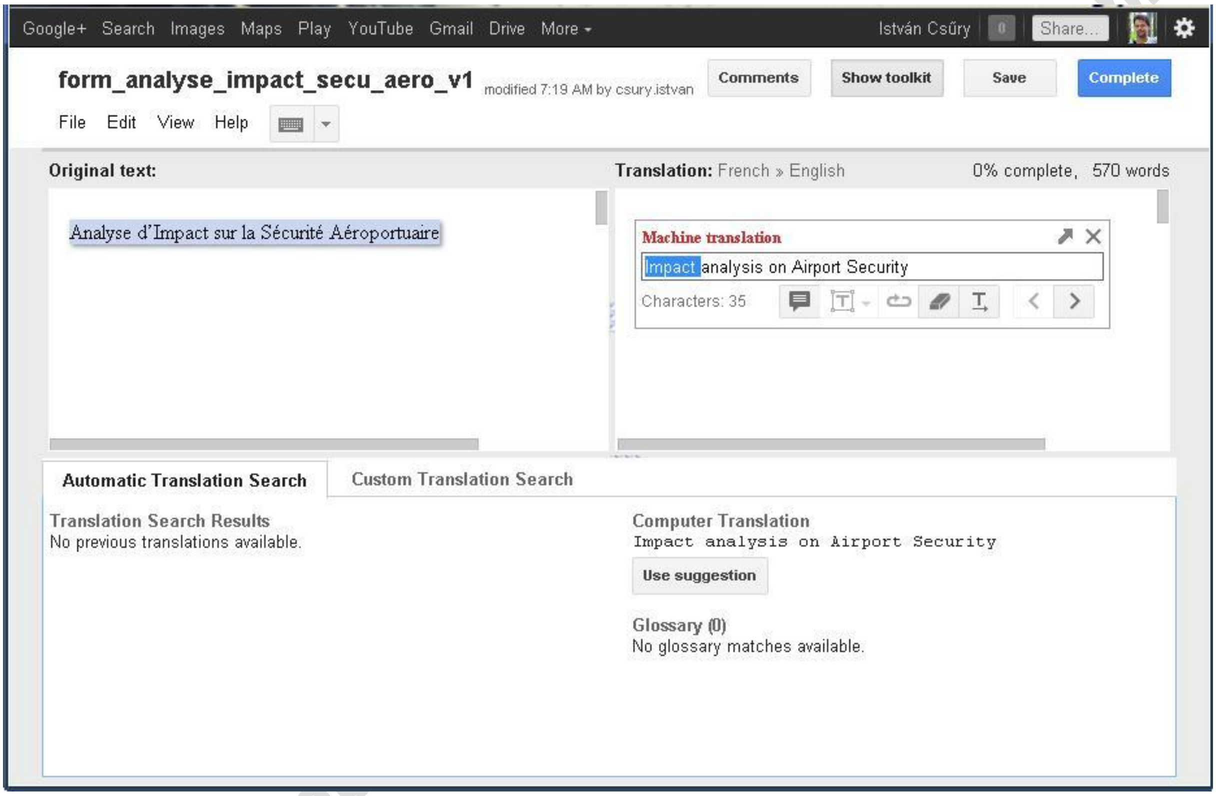
1. When opening the source text in your TM software, it will be automatically segmented before all. Segmentation will rely on formal cues like punctuation marks or paragraph marks. Of course, there may be mismatches between automatically created and senseful/translatable text units but you are allowed to manually shrink or expand any segment.
2. Segments of source text and those stored in the TM database are compared by the system. You may command the TM to auto-translate

segments in case of 100% match at any time during the translation process. If you want to verify the translation of every single segment (for example, when working with a legacy TM database in which you have not full confidence), you have the possibility of check and, if necessary, modify translations even of 100% matching segments.

3. Once the translation process started, you are presented with the segments to be translated one by one. If there is already a 100% or even a fuzzy (similar, but not identical) match in the TM database, it is displayed next to the source segment in an editable form that you can accept, modify or completely override. When you have finished with one segment, the system brings up the next for you, and it continues this way until you have reached the end of the source document.
4. Even if there is no somewhat matching segment in the legacy TM for the one you are translating, you may search in it for smaller parts of text, like a phrase or an expression. It helps very often to overcome particular difficulties of translation. In a similar way, when the TM software is integrated with a dictionary and/or a terminology database, you have as well facilitated vocabulary/terminology lookup functionalities at your disposal.
5. Everything you translate manually is automatically added to the TM so you can progressively take advantage of the work being already done if anything occurs repeatedly in the source text.
6. You may copy the content of a non-translatable segment from the source into the target text as well as any non-linguistic object.
7. At the end of the translation job, you may have several options: make an automatic quality check (if a reference database is provided for this purpose), save your work as is (for further operations, as proofreading, quality control or post-editing, carried out by someone else) or clean it up from special code (that the TM software might have added for technical reasons) and export it into the required document format. Note that using a TM software enables the translator to deal only with the content as the form is automatically taken care of. In other words, typography and page layout (as well as illustrations, etc) of the source document will normally be preserved.
8. A TM software enables TM database management as well. Users, for example, may apply settings to TM files that facilitate their proper use.

iii) Examples of TM / CAT software

The simplest example of CAT software is at your fingertips if you are a registered Google user. Besides its well-known general access online MT tool *Google Translate*, the company offers an online CAT application, too, named *Google Translator Toolkit* that provides a TM interface for translating documents in various file formats uploaded by the user as well as online content, like web pages or text copied and pasted into a text box. Users can pick an existing TM database or glossary or create one from scratch, and the system allows managing (uploading, downloading, deleting and sharing) these resources as well. Not only all basic TM functionalities are at our disposal but also we are presented with a machine-translated equivalent in target language of every segment of the source text. We can then accept, decline or modify it, and even add comments for later work phases. Collaborative translating is possible with other registered users of Google who may be granted to view, comment or edit our translation if we decide to share it. Here is a screenshot of the Google Translator Toolkit interface.



form_analyse_impact_secu_aero_v1 modified 7:19 AM by csury.istvan

Original text:

Analyse d'Impact sur la Sécurité Aéroportuaire

Translation: French » English

0% complete, 570 words

Machine translation

Impact analysis on Airport Security

Characters: 35

Translation Search Results

No previous translations available.

Computer Translation

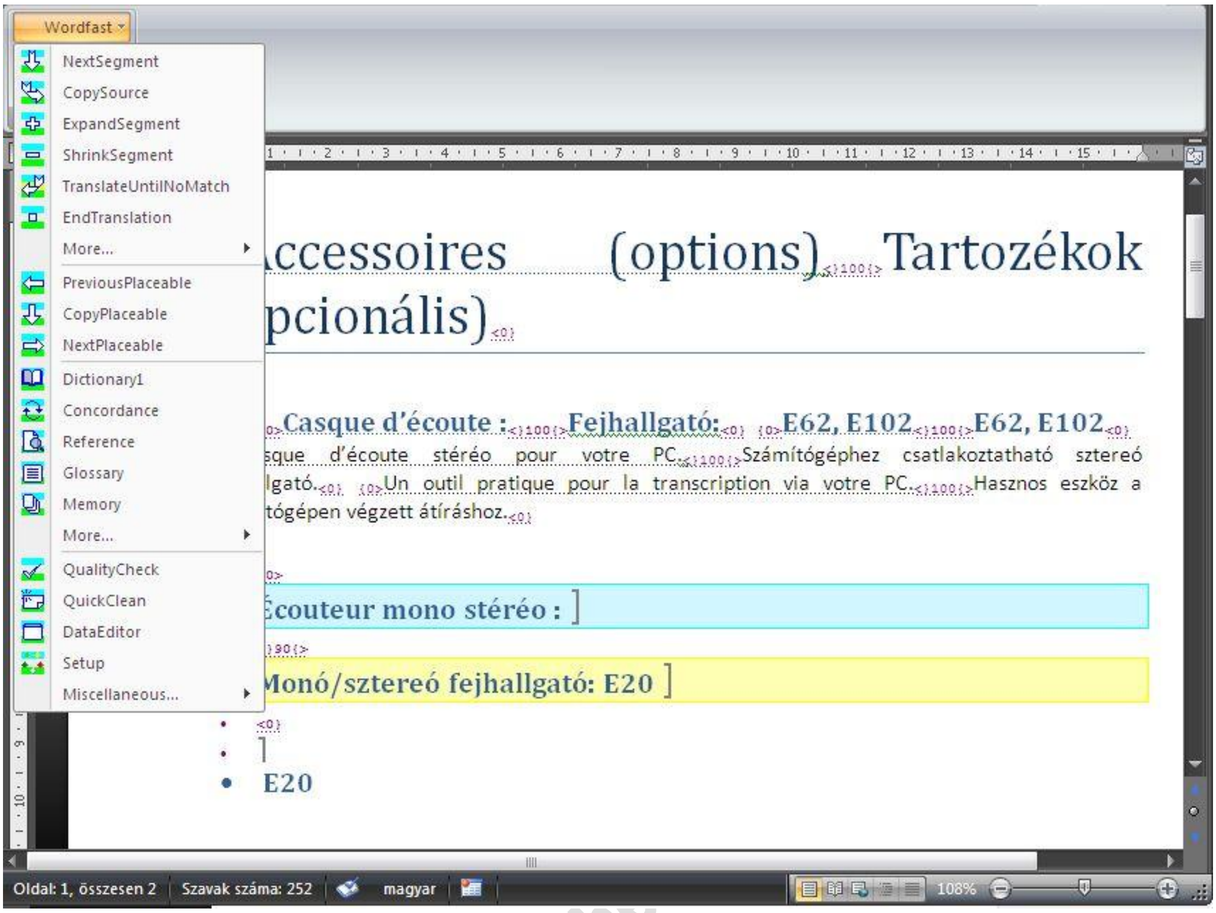
Impact analysis on Airport Security

Glossary (0)

No glossary matches available.

Figure 4 ***Google Translator Toolkit***

Figure 5 shows a MS Word window with the WordFast Classic TM plugin. The opened document contains segment boundary codes and the already translated segments in source (Hungarian) and target (French) language as well. Unnecessary elements will be cleaned up at the end of the translation process. The segment being translated is highlighted (in cyan); the translation may be edited in the lower box, which actually has a yellow background signifying that a partial match was found in the TM database and is displayed in the translation field for further editing.



ccessoires (options)<}100{>Tartozékok pcionális)<0}

0>Casque d'écoute :<}100{>Fejhallgató:<0} {0>E62, E102<}100{>E62, E102<0}

sque d'écoute stéréo pour votre PC<}100{>Számítógéphez csatlakoztatható sztereó lgató.<0} {0>Un outil pratique pour la transcription via votre PC.<}100{>Hasznos eszköz a tógépen végzett átíráshoz.<0}

0>

Écouteur mono stéréo :]

}90{>

Monó/sztereó fejhallgató: E20]

- <0}
-]
- E20

Figure 5 ***TM tool as a document editor plugin (Word Fast Classic)***

Evidently, many and more specific software are used in translation industry. In the present section, we are trying to provide some starting clues for the discovery of the contemporary translators' computerized equipment. The following list is, obviously, far from being exhaustive. Software being presented have also too many sophisticated features to be described here in the detail; web links are indicated for further information. Our table below presents only the most general and salient features highlighted by the software developers/vendors themselves. As we focus on the main functionalities of the tools, they might appear quite similar. However, considerable differences usually reside in the detail, not only concerning the features but also as a result of particular approaches and underlying "philosophies". In other words, two CAT software may seem equivalent with respect to WHAT they are good for but differ in HOW the same result is obtained. The choice depends on translators' preferences, needs, budget and working environment. While individual, free-lance translators usually equip themselves with some TM tool, companies and translation agencies implement a CAT/translation management system of their choice that translators working for them are required to use.

name of the tool	author / publisher / company; website	main features
SDL Trados Studio	SDL Language Technologies (SDL plc); http://www.sdl.com/cxc/language/translation-productivity/trados-studio/	• market-leading commercial software (free trial) • TM engine with expanded functionalities • connection possibility with MT and terminology tools • enhanced solutions for accelerating translation (e.g. sub-segment matching suggestions) • enhanced consistency and quality controlling tools • includes tools for revisions; enables to track, accept or deny changes • exporting / importing bilingual Word and Excel files • includes tools for managing the whole translation process • supports the most recognized industry standards (XLIFF – bilingual files, TMX – translation memories, TBX – terminology databases)
WordFast	Wordfast LLC; https://www.wordfast.net/	• three versions: a standalone application, a MS Word add-on and an online platform • commercial software but may be used in trial version with the only limitation that TM databases could not exceed 500 TUs • the online version is freeware • WF Professional: platform-independent • advanced time-saving features • simultaneous access to an unlimited number of TMs and glossaries with prioritizing possibilities • integration possibility with MT • real-time quality assurance • can handle a wide variety of file formats • widely used for training students in CAT

name of the tool	author / publisher / company; website	main features
memoQ	Kilgray; https://www.memoq.com	• three versions for individual use (including a free one) and two for teamwork • compatibility with other CAT tools (including those listed above) • enhanced TM features • many translation productivity boosters (e.g. automated concordance, predictive typing of auto-learnt expressions, automatic transformation of numbers-only segments in the target locale format, etc) • rich terminology tools • rich quality assurance features including real-time QA • integration possibility with market-leading MT systems • can handle a wide variety of file formats • project management features • reporting tools (e.g. homogeneity statistics in order to estimate how long the translation of a document will take) • customizability
Wordbee	Wordbee; http://www.wordbee.com/	• collaborative translation management system • although it contains a powerful (multi-lingual and multi-directional) TM tool, the system focuses on the overall process of translation service providing • complete translation project management solution including advanced document handling, communication facilities, cost calculation and invoicing as well • client and supplier management functionalities • business analytics features • integration possibility with some MT systems • handles a wide variety of file formats

3) Machine Translation

i) Beliefs and facts about MT

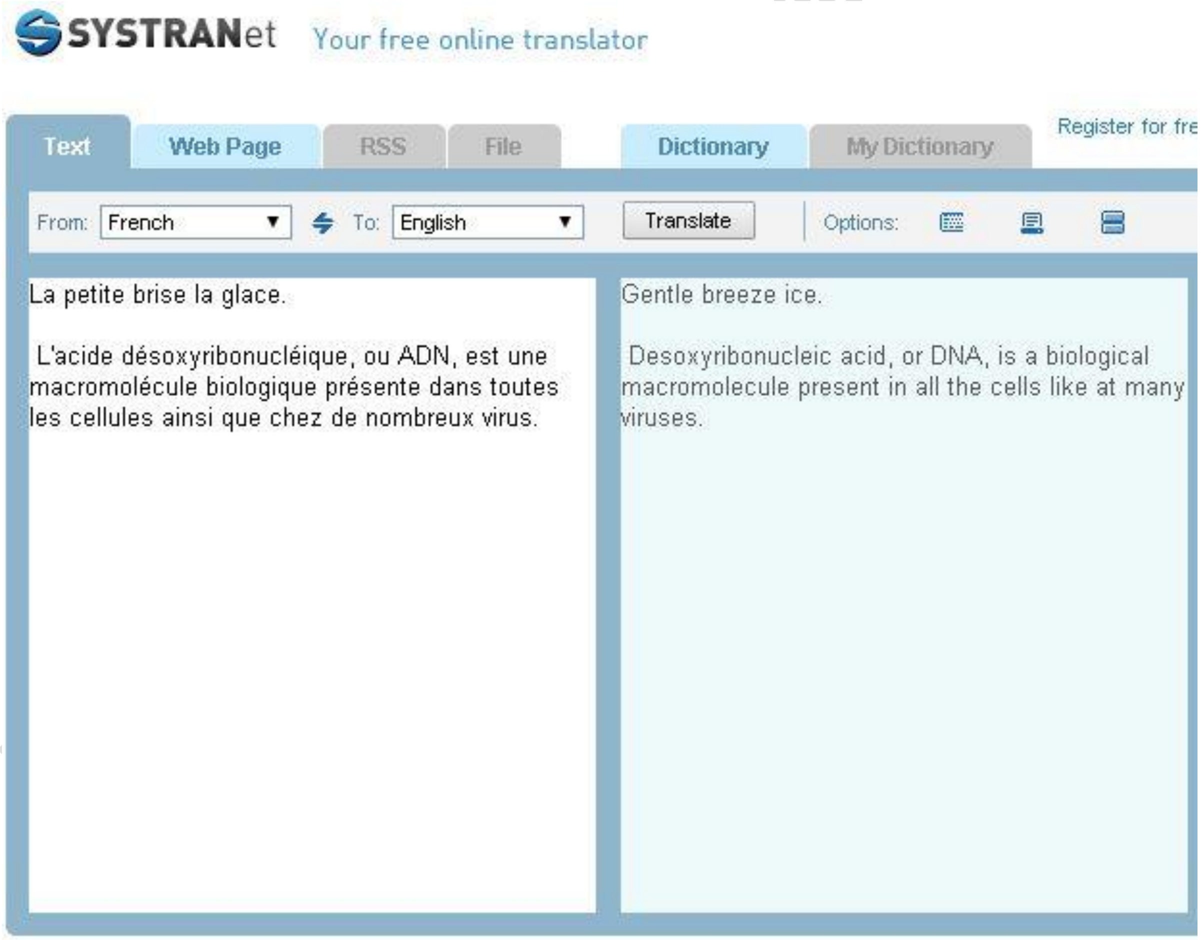
Experience average people have made with MT (or, more precisely, with some output of an MT system) leads them to exaggerated conclusions. Besides, several beliefs and half-truths circulate among the large public.

Some of these are related to the poor quality of many machine-translated texts that makes people perplex. All the same, we must not forget that the usefulness of MT as well as the quality of the output it is capable of are determined by some essential factors. First of all, MT is still not really all-purpose, or, at least, we have to adapt our expectations to the context in which we want to use it. On the one hand, MT performs poorly on thematically non restricted and linguistically uncontrolled texts, e.g. on a webpage with any content. However, that is the most known case, which gives rise to widespread disparagement of MT. What one should expect from this type of usage is not a human-like fully coherent translation containing mostly well-formed sentences but a series of raw equivalents of successive fragments of the original. This output is the best evaluated with regard to a total lack of comprehensibility for a reader who does not understand the source language at all. In this case, a very approximate automatic translation may be the unique way of getting an idea of the content of a document relevant to his interest. On the other hand, MT has proved to be efficient in special, more controlled contexts. If the subjects of the documents to be translated are well defined and, therefore, expressed with a more or less closed set of linguistic means (or, even more, if source texts are written with the idea of their automatic translatability in mind, perhaps in some controlled language), MT systems can produce good quality output, necessitating little post-editing. This type of MT use is solidly established and considered as an organic part of the digital working environment in translation industry, but it remains hidden from the large public. Needless to say, MT is not for literary translation, its incapacity to yield acceptable translations of literary texts is not a valid argument against its usefulness. (For examples, see the sources indicated below in the section *Further reading, sources/resources*.)

Before the age of smartphones and tablets, when PDAs and similar calculators with extended functions were in vogue, some of this kind of hardware contributed to the misbelief that MT is sold in form of machines or, at least, of as-is software like electronic dictionaries. In reality, one should prefer the term MT *system*, given its high level of complexity, and be aware of the considerable amount of effort to be invested in customizing it in order to make it really useful (Arnold & al. 1994: 11) for professional translation purposes. Despite all these considerations, our days' widespread applications turn our mobile phones or tablets in veritable MT devices if we can satisfy ourselves with rough translations of variable quality for a minimum of understanding in general purpose everyday use.

One should also have remarked that, even if perfection in this matter seems a very distant perspective, the performance of publicly available MT systems has nevertheless considerably improved. MT is a field of continuous research and development with an increasing rhythm of evolution, in spite of periods of decline having followed upswings of expectations and publicly funded projects. Even speech-to-speech MT, that seemed to fall under the heading of science fiction not very long ago, is becoming reality. Actually, availability and quality of MT is language-dependent: significant differences do exist not only following language pairs but also depending on the direction in which we want them to work.

Figure 6 shows an example of online MT illustrating what was said above. One of the two translated sentences, unlike the other one, presents multiple ambiguity issues in source language (French: ‘The little girl breaks the ice’ or ‘The gentle breeze freezes her’). Its English translation seems to confound its possible meanings and remains also fragmentary, whereas the second sentence, however far more long and complex, gives an acceptable result.



From: French ▼ To: English ▼

La petite brise la glace.

L'acide désoxyribonucléique, ou ADN, est une macromolécule biologique présente dans toutes les cellules ainsi que chez de nombreux virus.

Gentle breeze ice.

Desoxyribonucleic acid, or DNA, is a biological macromolecule present in all the cells like at many viruses.

Figure 6 *SYSTRANet at work*

ii) Approaches to MT

From the point of view of the language pairs: (historical) types of MT systems

Theoretically, the following types of MT systems can be conceived:

- bilingual systems (translating between a single pair of languages)
- multilingual systems (designed to translate among more than two languages)
- unidirectional systems (translating only from/to a single language to/from another one)
- bidirectional systems (translating in both directions in a given set of languages) (cf. Hutchins and Somers 1992: 69 sqq.)

However, actual solutions and the resulting typology of existing MT systems depend on other options as well, namely on the technological design of the system that may follow three basic strategies. (These approaches also correspond to a chronological order in the history of MT.) For a basic understanding of the possible procedures, let us represent translation schematically as a multi-phase process where the first one consists of the formal (lexico-grammatical) analysis of the source structures whereas in the last one, the forms of a semantically equivalent structure are generated in target language. Evidently, semantic interpretation and establishing semantic equivalence is far from being evident, so one might imagine one or more work phases accounting for this complex task as well.

Direct MT systems

First generation MT systems were founded on the (naive) hypothesis that source and target language structures could be directly linked as equivalents. Ultimately, this conception reduces MT to a kind of a sequential automated lookup of the words of a text in a bilingual dictionary, completed by some algorithms of morpho-lexical identification (in order to recognize inflected forms as the same lexical item) and surface adjustment for the output. Not surprisingly, this approach turned out soon to be inadequate.

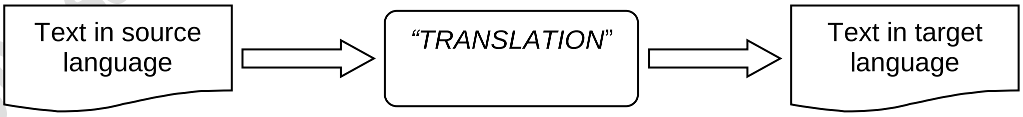
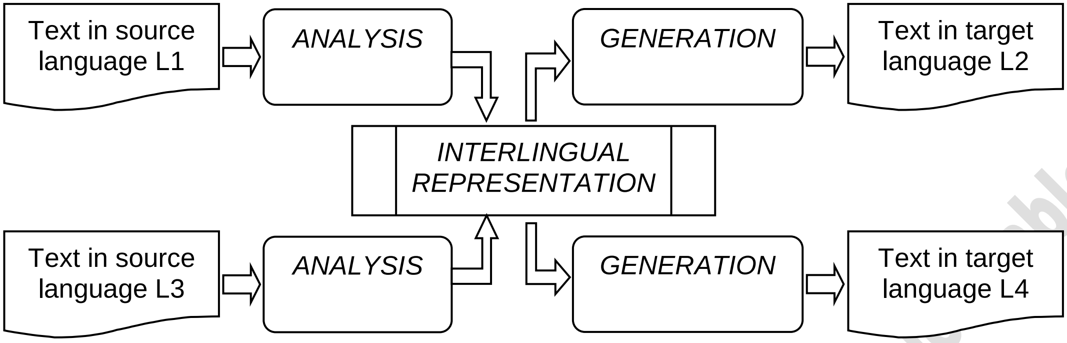


Figure 7 ***Direct MT***

Interlingual MT systems

Although there may be too much lexical mismatches between two languages to allow direct translation, information conveyed by a text is still transposable from one language in another. In other words, one should be able to realize MT by completing the automatic translation algorithms of analysis and synthesis with a language-independent representation of the semantic content of sentences. This abstract representation is called *interlingua* (literally: a "language between the languages"). In the interlingual approach of MT, the analysis of the target text is a more complex procedure as it has to yield an intermediary output (the abstract representation of the meaning of text segments). This will serve as input for the software module generating translated text in target language.

Figure 8 ***Interlingual MT***

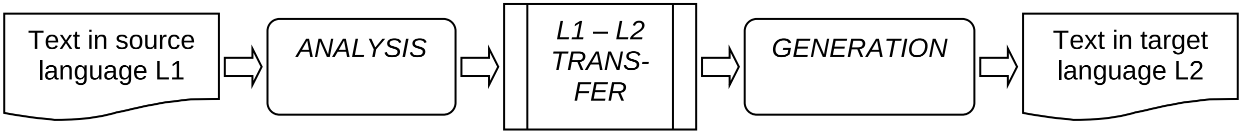

Text in source language L1
ANALYSIS
GENERATION
Text in target language L2
INTERLINGUAL REPRESENTATION
Text in source language L3
ANALYSIS
GENERATION
Text in target language L4

At first glance, it is a more realistic and realizable conception of (automated) translation. In addition, it should be more easy (and economic) to add new languages to an interlingual MT systems. Without engaging in a detailed discussion of the issues that, however, make it hard to turn it into practice, let us briefly note that interlingual MT as well is confronted with (at least three) major difficulties. The first one originates in the radical difference between natural and formal languages. In natural language use, ambiguity occurs frequently on the lexical as well as on the syntactic level. Humans deal with it quite well and, the more often, without realizing at all that they have something to disambiguate. As for MT development, computers need formal, non-ambiguous data and syntax. Conceiving rules for the disambiguation and semantic interpretation of texts in natural language is a difficult task, with unavoidable pitfalls and unforeseeable problems when the defined rules are applied in practice. What is more, not every piece of information in a text is explicitly coded as inference plays an essential role in human communication. Therefore, translators' brainwork may take into account extra-textual data potentially unavailable for a MT system. The second difficulty consists in the impossibility of developing a *really* language-neutral interlingua. Finally, one should not underestimate the impact of the costs of developing such systems (in terms of investment of efforts, time and money).

Transfer MT systems

The transfer approach to MT is similar to the interlingual one in the sense that translation is carried out via intermediate (abstract) representations between source and target languages, but it presents an essential difference inasmuch as these representations are not language-independent. The analysis module produces a representation specific to the source language. As the generation module needs a representation specific to the target language it can take as input, there is a bilingual transfer module linking together these two representations. This approach has the advantage of overcoming some difficulties of the interlingual one without losing its theoretically well-founded character and potential effectiveness.

Figure 9 Transfer MT



From the point of view of technology: two basic conceptions

Rule-based MT

The approaches described above (especially the interlingual and the transfer method) have in common that they are based on rules, which represent some kind of understanding of sentences they have to deal with. These rules are partly analogous to those of human language insofar as they combine a set of morpho-lexical items (a dictionary of words and inflections) with a set of syntactic rules (i.e. a grammar). At the same time, there are rules for disambiguation, identification and semantic interpretation as well as for translation. Rule-based MT systems present the following common characteristics:

- good quality translation of texts with a vocabulary and syntactic structures covered by the dictionary and the grammar rules implemented in the system,
- difficulty of handling unknown (or undisambiguable) structures,
- development of rule-based MT systems is more expensive and time-consuming.

Statistical MT

In a later phase of evolution, a radically different approach to MT has appeared and made spectacular progress. The method called statistical MT gives up making any attempt to “understand” what is to be translated. Statistical MT systems are similar to translation memory databases since they are based on huge bilingual parallel corpora, i.e. semantically related texts in two languages. The software is trained to automatically recognize (potentially) equivalent structures. When translating, the patterns found in source text are compared by the system to those contained in its database, and word sequences with a statistically higher probability in the given context are chosen among them. Statistical MT systems present the following common characteristics:

- variable quality translation: on the one hand, it depends on the characteristics of the source text, on the other hand, fluent-sounding sequences may be combined in barely, if at all, coherent units, which gives a fluctuating level of acceptability inside the same translated text;
- undisambiguable structures are easily translated (but not necessarily well),
- development of statistical MT systems is more rapid and less expensive.

It appears clearly that these two technological conceptions have comparative advantages and may be efficiently combined. As the hybrid approach to MT gains ground, rigid distinction between types of MT systems is tending to disappear.

iii) Examples of MT systems

There are many MT tools and services covering *variable language pairs* and designed for different frameworks, available from numerous providers. Occasional users will be (more or less) satisfied with free online solutions, like *Google Translate* (with 90 languages, https://translate.google.com), *Microsoft Translator* (with more than 45 languages, https://www.bing.com/translator/) or tools offered by *MorphoLogic* (with Hungarian and 12 other languages, http://www.webforditas.hu/). It is worth to compare more of them in practice. On these websites, we usually may enter (by typing or by copying and pasting) either some text or the address of a web page to be translated, or even upload a document.

MT systems for more demanding users range from individual mobile or desktop applications to highly customizable feature-rich company-level network solutions. Advances in statistical MT, substantial development of computer technology alongside with storage capacity as well as the growth of digitally available bi- and multilingual corpora has led to the apparition of new types of MT systems and services. For instance, instead of providing a ready-built MT tool for a given language pair (or a set of languages), some companies give us a framework for automatically building our MT engine adapted to the specific kind of texts we have to work with. For doing so, we need a sufficient amount of bilingual samples allowing to train the system (equipped with auto-learning algorithms).

As usually, the table below presents only a few examples of MT systems, intending to give the reader a hint of how much MT solutions may differ the ones from the others as for their approaches and the services they offer.

name of the tool	author / publisher / company; website	main features
SYSTRAN *(multiple products with different names)*	SYSTRAN International; http://www.systransoft.com/	• SYSTRAN is a market leading MT solutions provider, one of the oldest • originally rule-based, actually hybrid technology • offers more than 45 languages in more than 130 combinations (in SYSTRAN Enterprise Server 8) • MT tools for individual or corporate, mobile, desktop or online use

name of the tool	author / publisher / company; website	main features
Asia Online Language Studio	Asia Online Pte Ltd.; http://www.asiaonline.net/	• Asia Online is one of the youngest MT solutions provider • uses statistical MT but exploits clean data (data sourced from high-quality translations only) in order to obtain near-human publication quality • supports more than 530 language pairs (200+ language pairs under development)
KantanMT.com	Xcelerator Machine Translation Solutions Ltd.; https://www.kantanmt.com/	• leading cloud-based MT platform for the localization industry • uses statistical MT • allows to create customized MT engines using TM databases (i.e. subscribers have to feed their own data for training their own engine) • integrated applications including also data analytics and visualization and MT management software
Iconic Translation Machines	Iconic Translation Machines Ltd.; http://iconictranslation.com/	• system enabling to build not generic but domain-adapted MT systems (i.e. dealing with complex content in a specific field such as legal, pharmaceutical, and medical, that produce translations adapted to the language and style of the content and preserve the important information thanks to the “built-in” subject matter expertise) • immediate access to ready-to-use cloud-based fine tuned MT systems across a number of technical domains and languages • language-independent architecture but coverage is offered across 18 strategic language pairs

Further reading, sources/resources

Arnold, Douglas J. and Balkan, Lorna and Meijer, Siety and R. Lee Humphreys, and Sadler, Louisa (1994) *Machine Translation: an Introductory Guide*. London: NCC Blackwell. –

http://promethee.philo.ulg.ac.be/engdep1/download/bacIII/Arnold%20et%20al%20Machine%20Translation.pdf

Hutchins, W.John and Somers, Harold L. (1992) *An introduction to machine translation*. London: Academic Press. – http://www.hutchinsweb.me.uk/IntroMT-TOC.htm

Jurafsky, Daniel and Martin, James H. (2009) Speech and Langauge Processing. Pearson Prentice Hall.

Koehn, Philipp (2010) *Statistical Machine Translation*. Cambridge: Cambridge University Press. - http://www.statmt.org/book/

The Authors

ABUCZKI Ágnes – MTA–DE Research Group for Theoretical Linguistics

CSŰRY István – University of Debrecen, Department of French

ESFANDIARI Ghazaleh – University of Debrecen, Department of General and Applied Linguistics

FÖLDESI András – University of Debrecen, Department of General and Applied Linguistics

HUNYADI László – University of Debrecen, Department of General and Applied Linguistics

SZEKRÉNYES István – University of Debrecen, Department of General and Applied Linguistics