MITAO: A User Friendly and Modular Software for Topic Modelling

PAOLO FERRI
p.ferri@unibo.it
University of Bologna

IVAN HEIBI
ivan.heibi2@unibo.it
University of Bologna

LUCA PARESCHI
luca.pareschi@uniroma2.it
Tor Vergata University of Rome

SILVIO PERONI
silvio.peroni@unibo.it
University of Bologna

Abstract

Texts are among the most relevant data sources for social scientists, and traditionally researchers adopt qualitative methods for dealing with them. Yet, new computer aided techniques offer promising methodological avenues for scholars, which can now deal with big corpora of texts. One of the techniques that recently gained more relevance is Topic Modelling, which permits extracting bag of words which co-occur often in texts. While Topic Modelling was fruitfully used in sociology and management, existing software for performing it requires coding skills, and are not user friendly. In this paper we present MITAO, a new graphic-based, user friendly, open source software for performing topic modelling and other analysis on textual data.

1. Topic Modelling as a text analysis technique

Texts are one of the most relevant data sources for social scientists, and especially for those adopting qualitative research methods (e.g., Silverman, 2007; Flick, 2014). Traditionally, for interpreting texts and extracting meanings from textual sources, scholars rely on content analysis (Krippendorff, 2004): here data interpretation stems from an in depth reading of texts, which can be aided by the production of a set of themes guided by a research question and by prior assumptions by the researcher (DiMaggio *et al*., 2013). While these methods produced

wonderful and worthy research, they also have some limitations: dealing with epistemological issues (e.g., the relevance of intercoder reliability and the extent to which it is possible to have a real 'objective' stance toward data) goes far beyond the objectives of this paper, but it is worth nothing that traditional content analysis is unable to deal with big corpora of texts, and that the imposition of *a priori* categories can help the researcher in dealing with data, but at the same time presumes that the researcher knows what to search for in advance. Computer aided methods were therefore developed to help copying with these issues. The easier technique is a word count of some keywords within texts (Stone *et al*., 1966), but while this technique is widely used, relying on wordcount only infringes one of the most relevant principles of sociology, which is that the meaning of a word depends also on the surrounding words (DiMaggio *et al*., 2013). More sophisticated computer-based methods can better support research. In this paper we will thus focus on a software for textual analysis that, in its actual distribution, is able to perform Topic Modelling, a more sophisticated technique for dealing with texts and extract meanings from them.

Topic Modelling (TM) is a technique based on Bayesian statistic, and in particular, on an algorithm called Latent Dirilecht Allocation (LDA, Blei *et al*., 2003). Topic Modelling (TM) analyses texts and creates 'topics', which are bags of words that often co-occur together in the original texts (Mohr and Bogdanov, 2013). The underlying idea is that the algorithm is able to elicit a latent structure of topics, that constitute the texts in the corpus (Blei *et al*., 2003). As a result, all the words in the textual sources are coded to a topic, and all the original texts are constituted by the topic in different percentages. Topics can be ontologically different (Ferri *et al*., 2018), as they can be interpreted as themes, or discourse, or frames (Gamson, 1992). More in general, 'the sets of terms that constitute topics index discursive environments, or frames, that define patterns of association between a focal issue and other constructs' (DiMaggio *et al*., 2013: 563). DiMaggio *et al*. (2013) identify four relevant features of Topic Modelling: first, the results are explicit and the reproducibility of results is assured; second, the automated stage of TM permits dealing with big amounts of texts; third, at the same time, induction is just postponed to topics' interpretation, and this permits both discovering unexpected results, and using the same corpus of data for answering different research questions; finally, Topic Modelling is able to deal with *polysemy*, meaning that it is able to recognise that the meaning of a term depends on the surrounding words, and that the same word can have different meanings in different contexts.

In 2013, "Topic Models and the Cultural Sciences", a special issue of *Poetics* edited by John Mohr and Petko Bogdanov, paved the way for a wider use of Topic Modelling: here the technique was used to analyse newspapers' coverage of arts funding (DiMaggio *et al*, 2013), the different uses of language in academia (McFarland *et al*., 2013), the meanings regarding the nature of violence during the Qing Dinasty in China (Miller, 2013), the media and public attention to a terrorist alert (Bonilla and Grimmer, 2013), the 'grammar of motives' in national security strategic texts (Mohr and Bogdanov, 2013), the comparison of cross-national disciplinary evolutions of themes in texts (Marshall, 2013), the application of TM to humanities (Tangherlini and Leonard, 2013), and the analysis of the themes of about 3200 novel in 19th century novels (Jockers and Mimno, 2013). In management studies, Topic Modelling has been increasingly used for several purposes, such as detecting novelties and emergence, developing inductive classification systems, understanding online audiences, analysing frames and social movements, and understanding cultural dynamics (Hannigan *et al*., 2019). One example of the usage of Topic Modelling for detecting novelty and emergence of topics is the work by Ferri

et al. (2018), where TM was used to analyse twenty years of issues published by the journal *Accounting History* to analyse and measure the emergence or decline of topics, theories, and objects of study. Another example regards the use of TM to develop inductive classification systems, as it was used to study topic-based classification systems in patent data (e.g., Kaplan and Vakili, 2015); here Topic Modelling provided interesting results, as topic-based classifications do not perfectly overlap with formal systems of classification (Cho *et al*., 2017). As for understanding online audiences and products, in example, Wand and Chaundhry (2018) recently analysed online ranking for hotels and specifically focused on textual answers by managers to differently charged online reviews. As we already pointed out, topics can be deemed as frames (Goffman, 1974), and Topic Modelling can be very effective in providing an empirical methodology for eliciting frames: Levy and Franklin (2014), for example, adopted TM to inductively reconstruct frames use by individual and organisations in comments on a public website dealing with a political debate on the US truck industry. Finally, understanding cultural dynamics is particularly relevant for studies on institutional logics, where the impact of cultural meanings at field level is recognised (Thornton, Ocasio & Lounsbury, 2012), but it is hard to measure and specify them empirically (Hannigan *et al*., 2019). One example is the paper by Croidieu and Kim (2018), which used Topic Modelling and archival data analysis to study the emergence of U.S. wireless radio broadcasting field. More generally, as the language used in a document represents its cognitive content (Whorf, 1956), Topic Modelling is increasingly used in all those contexts where researchers are interested in constructing meanings starting with words (Hannigan *et al*., 2019).

In the next paragraph, we will briefly revise the existing software for performing topic modelling, and then we will move to present our software MITAO. MITAO aims at being an open source, user-friendly, modular, and flexible software for performing several kinds of text analysis. Here we present its architecture, composed by a *frontend* and a *backend*, both in its technical and substantial aspects. Then we proceed to explain how to perform Topic Modelling with MITAO, and we also provide a visual example. In the conclusion we reflect of the future development of MITAO.

2. How to perform Topic Modelling

A researcher has plenty of options for performing Topic Modelling. Yet, unluckily, all the available free software resources are not really user friendly, as they require programming skills, or the ability to understand programming languages to a certain extent. The state-of-the-art software for performing Topic Modelling is probably MALLET,[1] which is a Java based package for statistical natural language processing, document classification, clustering, topic modelling, information extraction, and other machine learning applications to text. MALLET is an open source software, developed within the University of Massachusetts Amherst and released under the Common Public License. In order to use MALLET, the researcher must use scripts in MS-DOS. Also, packages exist, within the environment R, to perform Topic

[1] Available online at http://mallet.cs.umass.edu/ (last accessed: February 20, 2020).

Modelling. Some examples are tm,[2] stm,[3] and lda.[4] Python libraries for working with texts, including Topic Modelling are Gensim[5] and Natural Language Toolkit (NLTK).[6] Finally, also David Blei's research group released code under several programming languages.[7] All these resources provide effective solutions for performing Topic Modelling, and few of the programs were coded by the same authors of the most relevant papers in the field. Yet, all these resources require programming skills which are not always available for social scientists. This is the reason that prompted us developing MITAO, which we will describe in the next paragraph.

3. MITAO

MITAO (Mashup Interface for Text Analysis Operations) aims at being an open source, user-friendly, modular, and flexible software for performing several kinds of text analysis. The current release of MITAO is able to convert data (from PDF to txt), clean data (stopword removal or removal of parts of text through the use of regular expressions), perform Topic Modelling, provide a quantitative measure of the results (trough perplexity score and topic coherence), visualise and save data. MITAO is a Python based web application, which could be installed on any operating system (e.g., Windows, MacOS, or Linux) and run locally on any modern web browser. It provides a human-friendly interface for creating a customisable visual workflow that can be shared among scientists for reproducibility purposes. While users without coding expertise deal with a user-friendly visual interface, Python developers can further customise and extend it, adding new features for handling other elaborations. The source code and documentation of MITAO is available on GitHub.[8] MITAO is currently licensed under the ISC License.

Considering the state of the art and the available solutions for performing Topic Modelling, users with no or poor skills in coding are likely to encounter problems when they try to: (a) use such methods for their own works using a particular programming language, or (b) describe the technical workflow of all their processing phases.

The main objective of MITAO is to overcome these limits and: (a) enable scientists to use text analysis operations, especially topic modelling, without having strong knowledge in all the technicalities of programming and software development, (b) build a comprehensive workflow containing text analysis operations, which could be shared with other colleagues as to provide the reproducibility of the results obtained.

---

[2] Available online at https://cran.r-project.org/web/packages/tm/index.html (last accessed: February 20, 2020).

[3] Available online at https://cran.r-project.org/web/packages/stm/index.html (last accessed: February 20, 2020).

[4] Available online at https://cran.r-project.org/web/packages/lda/index.html (last accessed: February 20, 2020).

[5] Available online at https://radimrehurek.com/gensim/ (last accessed: February 20, 2020).

[6] Available online at http://www.nltk.org (last accessed: February 20, 2020).

[7] Available online at http://www.cs.columbia.edu/~blei/topicmodeling_software.html (last accessed: February 20, 2020).

[8] Available online at https://github.com/catarsi/mitao (last accessed: February 20, 2020).

In this section, we will first talk about the architecture of MITAO and the conceptual model behind it. Next, we introduce a topic modelling use case, and illustrate MITAO's approach as to clarify its potentials.

3.1. Architecture of MITAO

Two are the levels used to represent the general architecture of MITAO as it is represented in Figure 3.1.1: (a) frontend, and (b) backend. The backend level is the core of MITAO and its definition and further customisation requires the expertise of a software developer. The frontend level is the part of MITAO which will be experienced by the final user. Here the features and operations exposed are easily usable without the need of any software programming skill. The frontend level does not have any effect on its underneath backend level, it only needs to have a reading access to the procedures defined in the backend level and convert/interpret such procedures into a usable dynamic interface.

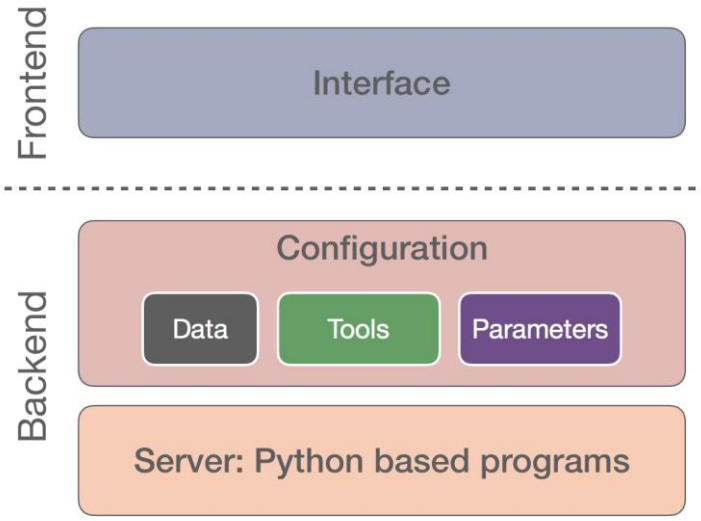

Figure 3.1.1. Architecture of MITAO.

3.1.1. Backend

The backend level contains two layers: (a) server, and (b) configuration. The configuration layer enables the definition of three types of entities to embed in MITAO: data, tools, and parameters. These entities are characterised through a set of attributes and functions which are implemented in the underneath layer (the server layer); therefore the two layers definition should be made parallelly: every attribute/function must have its corresponding implementation on the server layer.

The 'Data' entity represents a typology of data a user can work with and use in its MITAO workflow, (e.g., text files, PDFs, images, etc). These data typologies are also the ones a 'Tool' can produce on its output or can have as input. 'Tools' represent the operations MITAO is able to perform on data (e.g., Topic Modelling, data conversion, and data cleaning). 'Parameters' are needed by tools to perform their task. Obviously, each tool needs different parameters. The main idea of MITAO is to enable the final users to connect different 'Data' and 'Tool' entities

following the compatibilities between inputs and outputs. Users should also set the necessary attributes for the entities (e.g., 'data' or 'tool') they embed into the workflow. To let this happen, the configuration file should settle the 'Data' and 'Tools' entities with their corresponding 'Parameters' entities. Users have no control on the 'Data' entities which are produced as output of a 'Tool', therefore users cannot set the parameters of such 'Data' entities. The scheme of Figure 3.1.1.1 summarises the interconnections between these three entities.

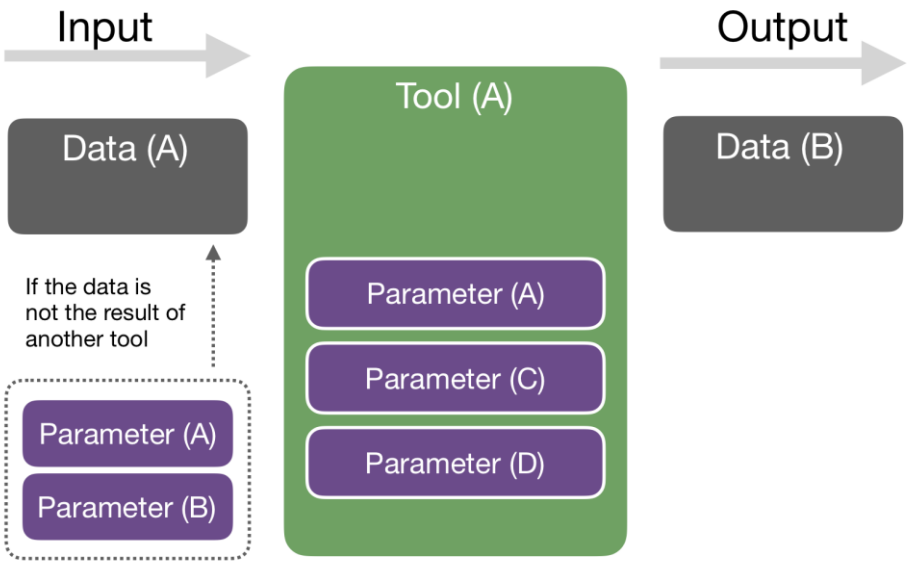

Figure 3.1.1.1. Logic connections between entities in MITAO.

The underlying idea of MITAO is that each of the three entities – 'Data', 'Tool' and 'Parameter' – covers a wide number of possible usages. 'Data' includes all the possible data sources, 'tool' all the possible operations, and 'parameters' all the necessary parameters. At the moment each new instance of these entities can be characterised as follows (see Figure 3.1.1.2):

- 'Tool': any instance of this entity can be part of the typology:
  - Filter: these tools take a 'Data' entity then perform some filtering operations on it and return the same 'Data' instance but filtered.
  - Text analysis: all the tools which perform text analysis operations are part of this instance. They take a 'Data' entity as input and return any other type of 'Data'.
  - Terminal: these tools have no output: their goal is to visualise (e.g., charts/diagrams) or store the input 'Data' entities.
- 'Data': any instance of this entity can be part of the typology:
  - PDF: document/s in .PDF format
  - Textual: document/s in .TXT format
  - Image: image/s in .PNG or .JPEG format

- Table: document/s which such that its inner information are organised in a field separated list table records (e.g., .CSV format).

- 'Parameters': any instance of this entity can have one of the following types, which represent the input typologies a MITAO user can make: (a) Free text, (b) List of options, (c) File, or (d) Checklist.

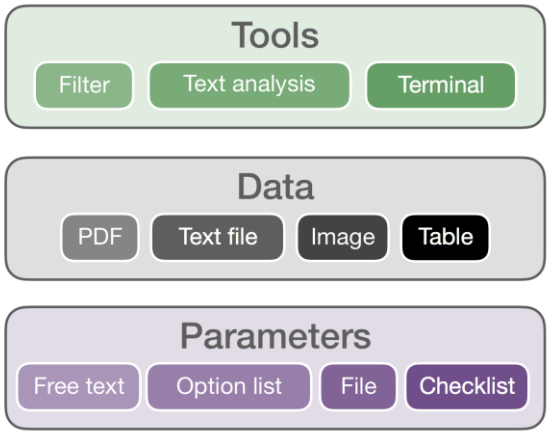

Figure 3.1.1.2. Available features for instances.

The configuration of MITAO should include all the three entities, and the relations between them. To do this MITAO provides one configuration file that embeds a code block for each possible instance of the three entities. The attributes and the relations with the other entities of each new instance are defined using a fixed set of fields. In Table 3.1.1.1, we list the three entities of the configuration layer and (a) give a brief description of it, (b) indicate the set of fixed fields needed to define it.

Entity	Definition	Configuration
Data	The data types a user can work with and let MITAO handle/understand.	ID: a unique identifier Title/Name: this value will be the entity's title/name in the interface. Type: the typology of the data (see Figure 3.1.1.2) Category: the data category (e.g., general input) Parameters: a list of 'Parameters' entities IDs
Tools	The tools a user can integrate in its workflow.	ID: a unique identifier Title/Name: this value will be the entity title/name in the interface. Type: the 'Tool' type (see Figure 3.1.1.2) Category: the data category (e.g., general input) Parameters: a list of 'Parameters' entities IDs

Entity	Definition	Configuration
		Compatible inputs: a list of 'Data' entities IDs which are accepted as inputs for this tool. *Output: a list of 'Data' entities IDs this tool generates as output. *in case the tool 'Type' equals 'Filter' or 'Text analysis'*
Parameters	The parameters a MITAO user can choose when adding a 'Data' or 'Tools' entities.	ID: a unique identifier Title/Name: this value will be the entity title/name in the interface. Type: the type of input handled the data (see Figure 3.1.1.2) *value: the value/s of the parameter init_value: the initial/default value *in case the parameter 'Type' equals 'option list' or 'Checklist' it should contain a list with all the values.*

Table 3.1.1.1. Entities, definition, and configurations.

3.1.2. Frontend

The frontend level of MITAO is defined on one interface layer which is based on the idea of having a dynamic graphical user interface (GUI) which interprets the functionalities of backend level into a comprehensive system for building and customising a complete text analysis workflow. Here we list the operations MITAO makes available to the users throughout its interface:

1. Adding and connecting 'Tool' and 'Data' entities in a directed graph network based diagram which represents the complete workflow;
2. Setting and editing the attributes of each workflow entity;
3. Running, saving and opening the workflows.

Considering the above operations, we designed a GUI as represented in Figure 3.1.2.1. The middle panel containing the workflow skeleton is based on a graph network schema, users have the ability to move and connect the inner nodes in order to create the final workflow to run. Along the above features, MITAO users should respect some rules/policies while using the interface:

- 'Tool' or 'Data' nodes can be connected to other 'Tool' nodes of the workflow only if their output is compatible with the input of the target 'Tool' node.
- MITAO will not allow users to create a directed circuit, cycle (the first and last vertices are repeated) in the network graph.

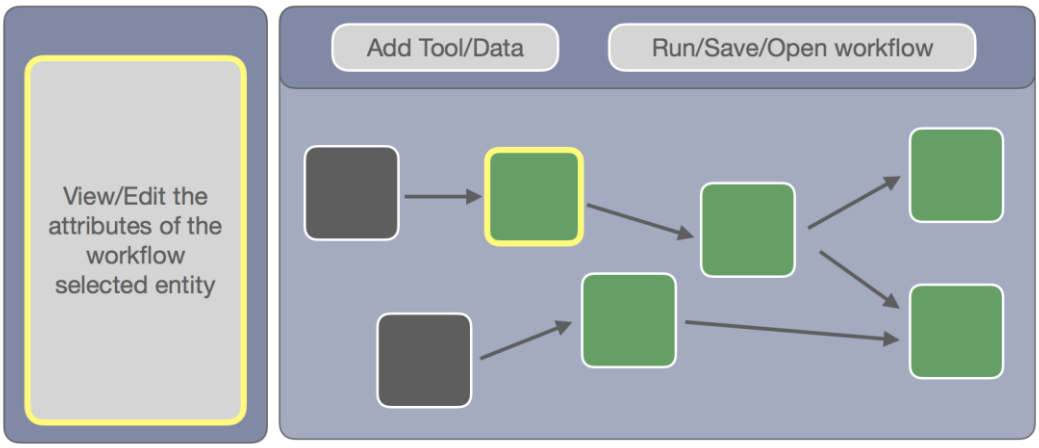

Figure 3.1.2.1. Scheme of the Graphic User Interface.

3.2. Technical aspects

The source code and documentation of MITAO is free and available on GitHub.[9] The backend level of MITAO is written in the Python programming language, and we used Flask,[10] a micro web framework written in Python to make it possible. On the other hand, the frontend of MITAO is web based and uses common programming languages used in web development, such as: HTML, CSS, Javascript, and jQuery. The only external library embedded is Bootstrap,[11] an open source toolkit holding common built-in web components.

To getting started with MITAO users have two available options:

1. The online version: MITAO is available at the address http://163.172.159.152:5000/. Users can access it from any common modern web browser, although we highly recommend using Chrome (since it was fully tested on it). The current server hosting MITAO has limited hardware potentials, therefore this option will allow limited operations: the maximum data limit users can upload is set on 2MB.
2. Run it on their local machine: MITAO can be installed on Windows, Mac, and Linux operating systems. Once installed it could run on any modern web browser, although once again we highly recommend using Chrome. In this case, users will have no restrictions on the operations and data payloads. The installation guideline is written in the MITAO documentation.[12]

MITAO interface is built as a one-page web application as in Figure 3.2.1 which highlights the interface operations, respecting the architectural model of MITAO (see previous section), so that the user can:

A. View/Edit the attributes of the workflow selected entity;
B. Add Tool/Data entities to the workflow;
C. See the workflow skeleton based on graph network schema;

[9] Available online at https://github.com/catarsi/mitao (last accessed: February 20, 2020).
[10] Available online at https://flask.palletsprojects.com/en/1.1.x/ (last accessed: February 20, 2020).
[11] Available online at https://getbootstrap.com/ (last accessed: February 20, 2020).
[12] http://ivanhb.it/src/mitao/mitao_doc.pdf

D. Run/Save/Open workflow.

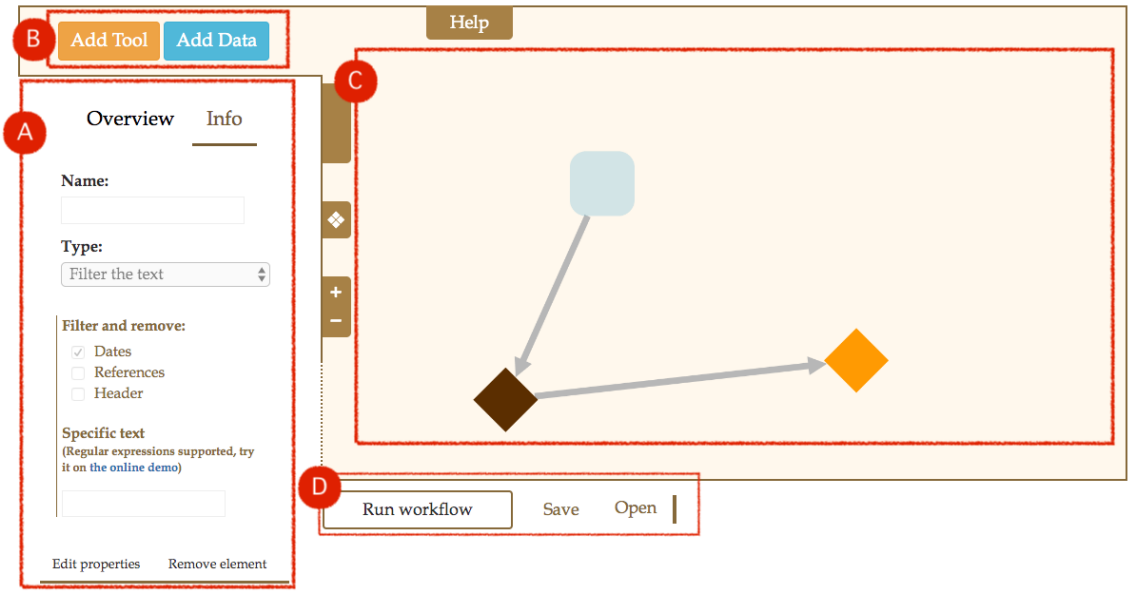

Figure 3.2.1. The web interface of MITAO.

4. Topic Modelling with MITAO: a use case

In this section we will show how to use MITAO in the definition of a complete workflow for making a topic modelling analysis. We will use the online interface and a small sample of seven accounting history papers. The selection of such a small sample is just for illustrative purposes, as Topic Modelling is able to deal with big corpora of texts. We want to visualise the final topics and save/download the results locally on our machine. To do so we will go through the following steps:

1. Add a 'Data' node and set the following attributes:
    a. Name: 'General Documents';
    b. Type: 'Textual document/s';
    c. File/s or Directory: select the directory where your .txt files are stored.

Figure 4.1 shows this entity added to the workflow, together with its attributes.

2. Add a 'Tool' node and set the following attributes:
    a. Name: 'LDA Topic Modeling';
    b. Type: 'Topic modeling with LDA';
    c. Stopwords language: 'English';
    d. Number of words: '10';
    e. Number of topics: '10'.

We asked the software to elicit 10 topics, and to show 10 words per topic for illustrative purposes. Also, after adding the 'tool' node, and setting its attributes, we selected 'General Documents', and connected it to 'LDA Topic Modeling'. Figure 4.2 shows the workflow after this step.

3. Add a 'Tool' node and set the following attributes:
    a. Name: 'Visualize Topics';
    b. Type: 'Visualize topics words'.

After adding 'Visualize Topics' we connected 'LDA Topic Modeling' to this new tool. Figure 4.3 shows the workflow after this step.

4. Add a 'Tool' node and set the following attributes:
    a. Name: 'Save LDA';
    b. Type: 'Save files'.

After adding 'Save LDA' we connected both 'LDA Topic Modeling' and 'Save LDA' to this new tool. Figure 4.4 shows the workflow after this step.

5. Run the built workflow and check its results.

Figure 4.5 shows the resulting workflow in MITAO after the execution of the process. Figure 4.6 shows what happens when clicking on 'show' after the execution of the process: here the 10 most important words for each topic are shown, and the dimension of each word accounts for the relevance of that word for the topic. Finally, clicking on 'save', a zipped file is downloaded, which contains (i) the list of the most important words for each topic (in a .csv file); (ii) a source per topic matrix detailing the topic composition of each source (in a .csv file); (iii) two files detailing perplexity score and coherence as measures of the quality of the model obtained (in .txt files); and (iv) the results of 'Visualize Topics' (in a .png file).

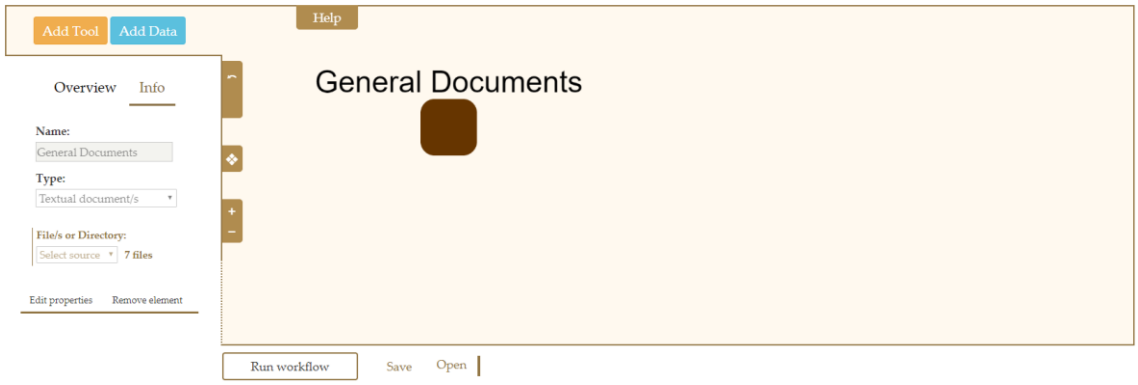

Figure 4.1. The data entity and its properties.

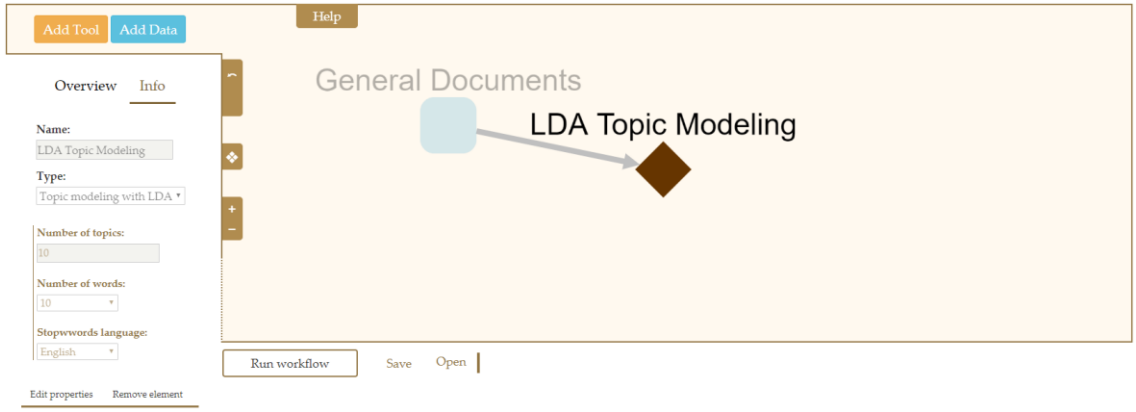

Figure 4.2. The LDA Topic Modeling Tool and its properties.

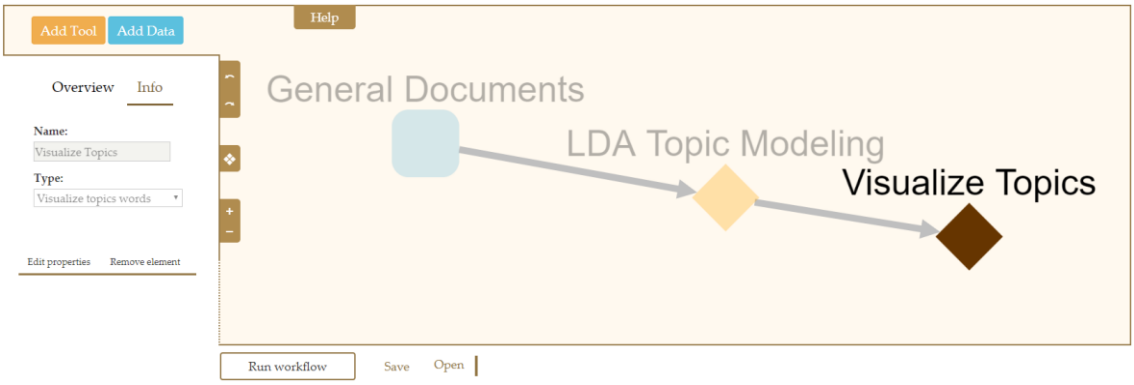

Figure 4.3. The Visualize Tool and its properties.

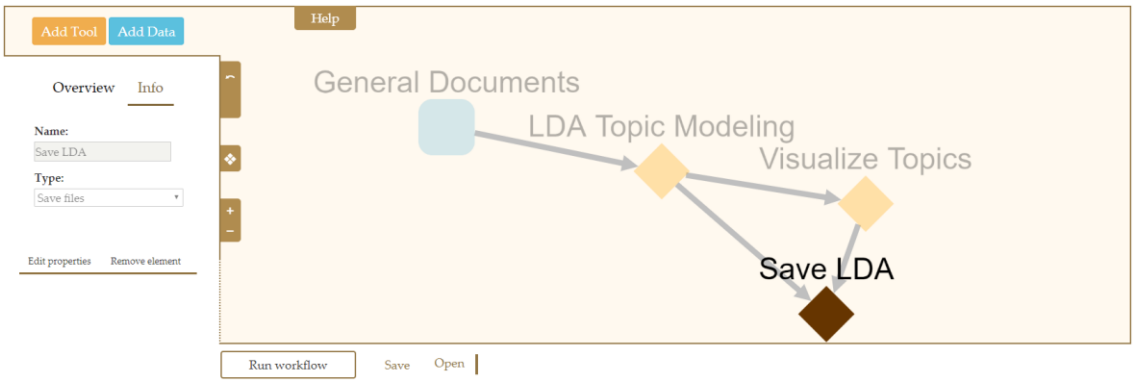

Figure 4.4. The Save tool and its properties.

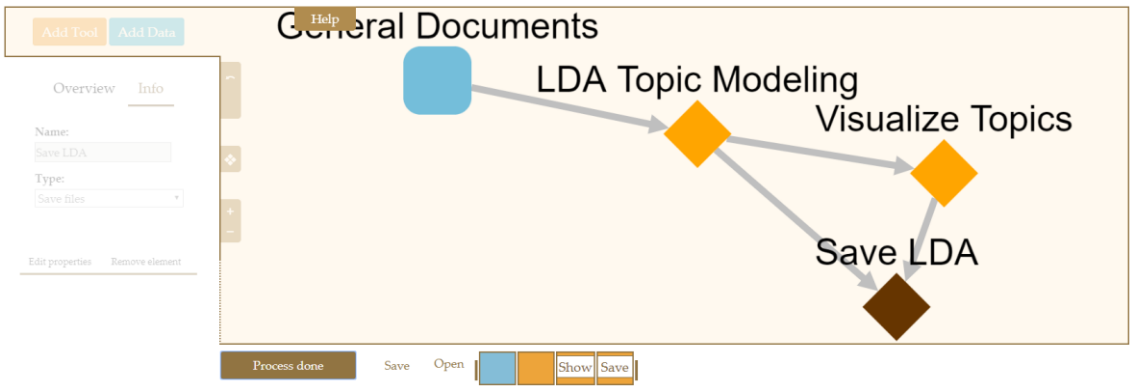

Figure 4.5. The workflow completed.

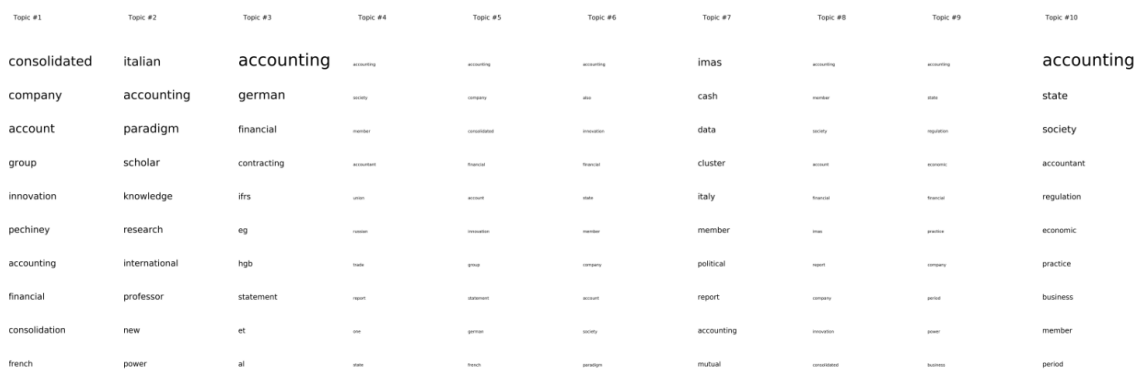

Figure 4.6. The visualisation of topics.

5. Conclusion

MITAO was developed within the CATARSI (*Comprensione Automatica di Testi e ARticoli nelle scienze Sociali e Informatiche* [automatic understanding of texts and articles in social sciences and computer sciences]) project funded by the University of Bologna under the program *Alma Idea 2017 – Linea Junior*. The aim of CATARSI is tackling the interface between social sciences and information science and improving both the knowledge and the development of computer-based techniques for analysing texts and extract meanings. The issue tackled by CATARSI is cultural and practical, and its results will impact both on information science, which deals more with ontological aspects, and on social sciences, which can use new instruments to improve the way knowledge is analysed and created. MITAO is the first result in terms of instruments, as it permits social scientists to easily perform Topic Modelling. As we already described, the current release of MITAO is able to convert data (from PDF to txt), clean data (stopwords removal or removal of parts of text through the use of regular expressions), perform Topic Modelling, provide a quantitative measure of the results (trough perplexity score and topic coherence), and visualise and save data. Nonetheless, MITAO has some limitations, which can be defined as tool yet to be implemented. Indeed, MITAO was designed with a modular structure that easily permits introducing new tools and new kinds of data. We are already planning new versions of MITAO that will introduce tools in the pre-processing of data, in the analysis, and in the visualisation phase. As for the pre-processing, we aim at introducing different kinds of tools to prepare data before the analysis. For example, we want the user of MITAO to be able to (i) lemmatise and stem words to transform words into their roots; and to (ii) instruct the software with the use of bi-grams or n-grams, which are words that must be analysed together to retain their meaning (e.g., 'corporate social responsibility', or 'publishing house'). Data preparation is a relevant step for textual analysis, and the introduction of these features will permit a better customisation of the analysis. As for the analysis, we want to add to MITAO some tools which are already present in NLTK but need 'translation' to be used in our user-friendly interface. Some examples include named entity recognition, part-of-speech tagging, semantic network analysis, and sentiment analysis. Named entity recognition searches through unstructured texts for entities pertaining to predefined categories, such as person names, organisations, and geographical locations. Adding these features will permit the user of the software to automatically find the 'entities' (persons, places, institutions, etc.) mentioned in the text. Part-of-speech tagging defines the process of classifying words into their parts of speech, thus labelling them accordingly, and

permits more nuanced analysis on the actions described in the sentences. Semantic network analysis focuses on the structure of words within texts with methods of network analysis to gain insights regarding the semantic relations among concepts. Finally, sentiment analysis uses dictionaries to evaluate the charge of words within given texts (e.g., positive, neutral, negative), thus characterising those texts. By introducing these features, we will be able to describe relationships among words and define a positive/neutral/negative loading for topics. As for the visualisation, we are adding a Python library for interactive topic model visualisation, which is called pyLDAvis.[13] This library is intended to help users interpret the output of a topic modelling, which is the topics fit to a corpus of text data. The package extracts information from a fitted LDA topic model to create an interactive web-based visualisation. This visualisation permits exploring topics focusing on two kinds of words: first, on the words with higher prevalence within the topic, if compared to the whole corpus, which is a normal output of TM. But, second, it permits also discovering rare words which highly characterise each topic.

While work is in progress for the future releases of MITAO, a previous version was already presented and tested during a symposium organised within the EURAM 2019 Conference – Exploring the future of management, which took place in Lisbon (Portugal) in June 2019. We proposed an interactive workshop, where 10 participants had the chance to test a beta version of MITAO. In particular, we offered participants few datasets different for source and type of file, and scholars played with our software for analysing data. They tested the different features of MITAO and highlighted criticalities, but also suggested avenues for improving the software both regarding its usability, and the features offered. Our aim is to keep improving MITAO, while also offering it to the scientific community for testing and using the software.

Keywords

text analysis; discourse analysis; topics; frames; discourse; themes; topic modelling; visualisation; MITAO; Python

Reference list

Blei, D.M., Ng, A.Y., and Jordan, M.I. (2003), "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, 3 (1): 993–1022.

Bonilla, T., and Grimmer, J. (2013), "Elevated Threat Levels and Decreased Expectations: How Democracy Handles Terrorist Threats", *Poetics*, 41 (6): 650–669.

Cho, Y.-J., Fu, P.-W., and Wu, C.-C. (2017), "Popular Research Topics in Marketing Journals, 1995–2014", *Journal of Interactive Marketing*, 40: 52–72.

Croidieu, G., and Kim, P.H. (2018), "Labor of Love: Amateurs and Lay-Expertise Legitimation in the Early U.S. Radio Field", *Administrative Science Quarterly*, 63 (1): 1–42.

DiMaggio, P., Nag, M., and Blei., D. (2013), "Exploiting Affinities between Topic Modeling and the Sociological Perspective on Culture: Application to Newspaper Coverage of U.S. Government Arts Funding", *Poetics*, 41 (6): 570–606.

[13] Available online at https://pypi.org/project/pyLDAvis/ (last accessed: February 20, 2020).

Ferri, P., Lusiani, M., and Pareschi, L. (2018), "Accounting for Accounting History: A Topic Modeling Approach (1996–2015)", *Accounting History*, 23 (1-2): 173–205.

Flick, U. (2014), *An Introduction to Qualitative Research*, London: SAGE.

Gamson, W.A. (1992), *Talking Politics*, Cambridge: Cambridge University Press.

Goffman, E. (1974), *Frame Analysis: An Essay on the Organization of Experience*, Cambridge, MA: Harvard University Press.

Hannigan, T., Haans, R.F.J., Vakili, K., Tchalian, H., Glaser, V., Wang, M., Kaplan, S., and Devereaux Jennings, P. (2019), "Topic Modeling in Management Research: Rendering New Theory from Textual Data", *Academy of Management Annals*, 13 (2): 586–632. DOI: 10.5465/annals.2017.0099.

Jockers, M.L., and Mimno, D. (2013), "Significant Themes in 19th-Century Literature", *Poetics*, 41 (6): 750–769.

Kaplan, S., and Vakili, K. (2015), "The Double-Edged Word of Recombination in Breakthrough Innovation", *Strategic Management Journal*, 36 (10): 1435–1457.

Krippendorff, K. (2004), *Content Analysis: An Introduction to Its Methodology* (2nd edn.), Thousand Oaks, CA: SAGE.

Levy, K.E.C., and Franklin, M. (2014), "Driving Regulation: Using Topic Models to Examine Political Contention in the U.S. Trucking Industry", *Social Science Computer Review*, 32 (2): 182–194.

Marshall, E.A. (2013), "Defining Population Problems: Using Topic Models for Cross-National Comparison of Disciplinary Development", *Poetics*, 41 (6): 701–724.

McFarland, D.A., Ramage, D., Chuang, J., Heer, J., Manning, C.D., and Jurafsky, D. (2013), "Differentiating Language Usage through Topic Models", *Poetics*, 41 (6): 607–625.

Miller, I.M. (2013), "Rebellion, Crime and Violence in Qing China, 1722–1911: A Topic Modeling Approach", *Poetics*, 41 (6): 626–649.

Mohr, J.W., and Bogdanov, P. (2013), "Introduction – Topic Models: What They Are and Why They Matter", *Poetics*, 41 (6): 545–569.

Silverman, D. (2007), *Interpreting Qualitative Data*, London: SAGE.

Tangherlini, T.R., and Leonard, P. (2013), "Trawling in the Sea of the Great Unread: Sub-Corpus Topic Modeling and Humanities Research", *Poetics*, 41 (6): 725–749.

Thornton, P.H., Ocasio, W., and Lounsbury, M. (2012), *The Institutional Logics Perspective: A New Approach to Culture, Structure and Process*, New York, NY: Oxford University Press.

Wang, Y., and Chaudhry, A. (2018), "When and How Managers' Responses to Online Reviews Affect Subsequent Reviews", *Journal of Marketing Research*, 55 (2): 163–177.

Whorf, B.L. (1956), *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf*, Cambridge, MA: Technology Press of Massachusetts Institute of Technology.