ITMT: Interactive Topic Model Trainer

L. Calvo-Bartolomé, J. A. Espinosa-Melchor, and J. Arenas-García
Universidad Carlos III de Madrid
Avda. Universidad, 30. 28911 Leganés, Madrid, Spain
lcalvo@pa.uc3m.es
joespino@pa.uc3m.es
jarenas@ing.uc3m.es

Abstract

Topic Modeling is a commonly used technique for analyzing unstructured data in various fields, but achieving accurate results and useful models can be challenging, especially for domain experts who lack the knowledge needed to optimize the parameters required by this natural language processing technique. From this perspective, we introduce an Interactive Topic Model Trainer (ITMT) developed within the EU-funded project IntelComp. ITMT is a user-in-the-loop topic modeling tool presented with a graphical user interface that allows the training and curation of different state-of-the-art topic extraction libraries, including some recent neural-based methods, oriented toward the usage by domain experts. This paper reviews ITMT's functionalities and key implementation aspects in this paper, including a comparison with other tools for topic modeling analysis.

1 Introduction

In the growing information age, today mostly dominated by an unprecedented interest in artificial intelligence (AI), as well as its deployment in a multitude of applications, topic modeling is still mostly preferred over other AI techniques for the automatic extraction of the main themes concurring in a collection of documents.

Nonetheless, the blind application of these topic extraction tools entails some difficulties, from which we can cite the presence of garbage topics (i.e., topics that describe the corpus under analysis as a whole, but not the relevant topics it consists of); the complicated adjustment of flat topic models when the corpus is characterized by topics with very different sizes, as they do not support hierarchical modeling; or challenges associated with finding a suitable tuning for each algorithm, which requires expertise and a good knowledge of their hyperparameters, just to mention some.

Moreover, when the knowledge of domain experts is available, it is worthwhile to offer tools that enable the incorporation of such understanding into the building of topic models, providing both appliances for visualization and guided adjustment of the model, specially designed for the usage of non-AI practitioners that are experts in their area. Nevertheless, it is essential that the models created for this purpose are easily interpretable by end users, i.e., avoid garbage or too broad topics, etc.

Hence, we present in this paper IntelComp's *Interactive Topic Model Trainer (ITMT)*, a tool developed within the *H2020 European project IntelComp*[1] for this purpose. IntelComp seeks the development of a platform that makes use of the latest generation of Artificial Intelligence and Natural Language Processing (NLP) tools to provide relevant information to assist public policies in Science, Technology, and Innovation (STI), geared toward aiding decision-making over the policy cycle. This requires a thorough analysis of documentary sources, which can entail up to hundreds of millions of documents (e.g., scientific articles, patents, etc.); therefore, here becomes fundamental the use of topic modeling to extract information with a level of detail greater than attainable through the inspection of these sources' metadata.

ITMT consists of a Python-based toolbox integrated within a PyQT6-based graphical user interface[2] for the training of topic models following an expert-in-the-loop approach that ultimately contributes to models that are more aligned with the prior experience and needs of IntelComp end users. The software package includes several state-of-the-art topic modeling solutions, seeking adequacy to the needs of each possible scenario, due to both the characteristics of the data sets and the scalability of the algorithms, but also the infrastructure available for training. Besides, the software contains a series of proprietary algorithmic improvements that allow

[1] http://intelcomp.eu

[2] The project will also make the tool accessible via a web service.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics
System Demonstrations, pages 43–49

Features	ITMT	Gensim	Mallet	StandfordTMT	STTM	Familia	TopicNet	ToModAPI	OCTIS	BigML
Pre-processing tools	✓	✓	✓	✓				✓	✓	✓
Bayesian based models	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Neural topic models	✓							✓	✓	
Level-2 HTMs for granularity inspection	✓									
Topic visualization tools	✓						✓		✓	✓
Topic curation tools	✓									
Topic annotation tools	✓									
Hyper-parameters tuning		✓	✓	✓		✓			✓	
Global topic statistics	✓			✓						
Coherence metrics	✓	✓	✓		✓		✓	✓	✓	
Entropy metrics	✓									
Diversity metrics									✓	
Significance metrics									✓	
Classification metrics			✓		✓			✓	✓	
Domain-experts oriented	✓					✓	✓			

Table 1: Comparison of ITMT with other existing frameworks for topic modeling.

topic models to be evaluated and curated by experts.

Compared to other current topic modeling frameworks (McCallum, 2002; Rehurek and Sojka, 2010; Lisena et al., 2020; Silvia Terragni, et al., 2021), which typically focus on putting out topic modeling algorithms but ignore their interpretability and adequateness for the needs of end users, ITMT stands out as a tool for training topic models while including the knowledge of experts in the creation and curation of such models. A comparison summary between ITMT and other available frameworks is available in Table 1, while a detailed analysis is provided in Section 4.

The main contributions of our framework's current release are:

- Integration of several topic extraction libraries enabling users to easily train models under a common interface.
- Incorporation of a novel implementation of Hierarchical Topic Models (HTMs). In particular, we provide a level-2 HTM comprising tools that allow the user to pick which topics should be further split.
- Inclusion of topic evaluation, annotation, visualization, and curation tools aiming for the usage of domain experts, which are common and independent of the training algorithm.

ITMT has been published under a permissive MIT license in the GitHub Project `https://github.com/IntelCompH2020/topicmodeler`.

2 System overview

ITMT consists both of a PyQT6-based graphical user interface (GUI) as the front-end and a back-end service supporting all the operations that need to be carried out as a response to user interactions. The visualization itself and the actual training and optimization of the models are completely decoupled. The state management is performed on the back-end side, and it is sustained by the use of external folders given as input to the application, as described in Section 2.1. This provides the ITMT with both persistence and portability capabilities, as all structures (training datasets, models, etc.) created during the application's execution can be accessed and modified at a later time.

2.1 Input requirements

For the system to work, three input folders must be provided, namely 1) a *project folder* in which the application's output will be saved; 2) a *source folder* containing the datasets; 3) a *wordlists folder* to harbor the wordlists (i.e., lists created by the user outside the ITMT and the ones generated during the application's execution). For the time being, the datasets available in the source folder must be given in parquet format, and contain at least the raw version of the texts to be used for training, and, for some models, their contextualized embeddings.

Provided the three inputs, the project folder is set up with a fixed structure composed of 1) a configuration file with all the specifications and descriptions of variables implied in the ITMT; 2) a folder for the training datasets, and 3) a folder for the trained topic models.

2.2 The graphical user interface

So as to offer the distinct ITMT utilities in a user-friendly manner, the GUI is composed of four main subwindows, each of them relating to one of the functionalities offered by the application, leading the user through the different steps that must be followed for the creation of topic models; and one additional subwindow serving as a welcome page.

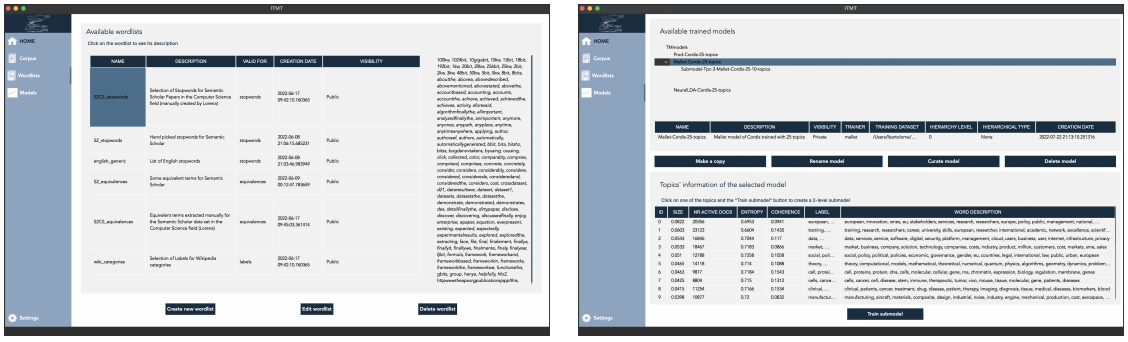

Figure 1: Wordlist management and Model management ITMT's subwindows. Left: Available wordlists and their information are listed, and the user can access their content by clicking on a wordlist's associated row; also, the user can access the menus for editing and creating a new wordlist, or delete any of them. Right: Available models (level-1 and level-2) are hierarchically listed; by clicking on any, the model's information, topical description, and metrics are displayed; windows for curation and training of submodels can be accessed from here.

In the following, we offer a summarized description of each of these subwindows, and for more details, the demo video available here[3] can be consulted.

Welcome page. It allows the selection of the project/source/wordlists folders through either the user's file system or a list of recently used folders and provides a shortcut and description to all functionalities.

Corpus management. It is composed of two views, each of them serving a different purpose: 1) visualizing and operating with the available local datasets (obtained from the source folder) and 2) visualizing and operating with the user-created training datasets. From 1) a training dataset can be created through a new window and from 2) the preprocessing + topic modeling training windows can be accessed.

Wordlist management. It supports the listing, creation, edition and deletion of *ad-hoc* word lists to incorporate information collected by domain experts (e.g., stopwords, equivalent terms or acronyms, etc.).

Model management. It assists the models' management functionalities (listing, copying, renaming, and deletion), as well as the visualization of the models' information and statistics. Also, a thematic analysis with different levels of resolution through the construction of second-level topic models can be pursued from here, and the curation, visualization and evaluation tools can be accessed.

Settings. It permits the user to configure all the settings available in the configuration file.

[3] https://youtu.be/e0YDsnnNHto

2.3 ITMT users' workflows

With the ITMT's subwindow division, we aim at guiding the user through the steps that are necessary for the procurement of high-quality models with interpretable topics, and as aligned as possible to the needs of the expert orchestrating the creation. To do so, we recommend following the succeeding five-stage process, summarized in Figure 2. Note that the tool itself does not impose the execution of this workflow, but it is the user who should be conscious of it.

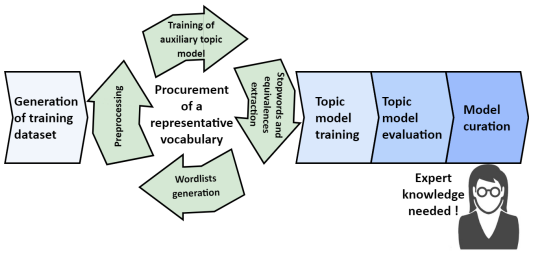

Figure 2: ITMT recommended workflow.

1. **Generation of a training dataset.** By utilizing the documents of one of the local datasets provided in the source folder, or the concatenation of several of them, the user can construct a training corpus through the Corpus management subwindow.

2. **Procurement of a representative vocabulary for the training corpus.** After the selection of the just created training dataset, its preprocessing and the creation of an initial auxiliary model with a moderate number of topics (e.g., 30-40 topics) through the preprocessing and training windows should be approached. Having the model constructed, the user benefits

from some topic evaluation and curation tools to clean the vocabulary as follows:

- Through the visual and manual inspection of the model, garbage topics can be identified. This allows the detection of uninformative terms for the corpus, which should be marked as stopwords.
- Terms that appear in several topics of varied nature should also be marked as stopwords.
- Equivalent terms coming from lemmatization errors, acronyms, and synonyms should be marked to be mapped to a common structure.

Based on the stopwords and equivalences detected, corresponding wordlists for filtering each of the latter can be created through the Wordlist management subwindow, which can be used for performing a new preprocessing of the training corpus. This process can be repeated any number of times until an adequate vocabulary for the dataset is obtained.

3. **Topic modeling training.** Having a representative vocabulary, the final training should be pursued. To obtain an easily interpretable model, slightly overestimating the number of training topics is a good practice.

4. **Topic modeling evaluation.** It could be advantageous for the user to train several models and then pick the one with the best performance metrics.

5. **Models curation.** Finally, the usage of curation tools is recommended for the final adjustment of the selected model. E.g., similar topics could be fused into a unique one or garbage topics removed; alternatively, the user can observe the presence of too broad topics, for which a level-2 exploration through the HTM techniques may be of use, etc.

3 Software components

As we have covered in the former section, the construction of a topic model is a procedure that requires the sequential execution of various tasks, each of them managed by a different component of the topic modeling service underlying the GUI. We present in this section each of these components.

3.1 Preprocessing pipeline

This section describes the tasks carried out by the ITMT's preprocessing pipeline. It is important to highlight it is not a complete NLP pipeline but only provides additional cleaning tasks usually recommended to obtain higher-quality topic models.

The transformations to which the documents inputted into the pipeline are subjected are based on a set of settings selected *ad-hoc* by the user, which includes the wordlists to be used for the vocabulary cleaning (i.e., steps 1 and 2 described below) and the parameters implied in steps 3 and 4 (e.g., vocabulary size, etc.).

1. **Removal of additional stopwords** which, while having meaning to a sentence, lack semantic interest for the dataset under analysis.

2. **Word replacements** by other equivalent ones so that they are treated as a single term during topic modeling.

3. **Filtering of short documents**, as they lack enough information for a robust estimation of their thematic composition.

4. **Vocabulary construction** by removing terms with a too high or too low probability of appearance in the corpus, and restricting the maximum vocabulary size.

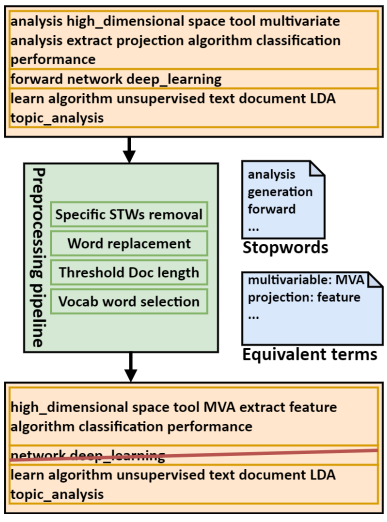

Figure 3: Illustration of the operation of the components in the ITMT preprocessing pipeline.

The output of the processing pipeline is therefore a subset of the input documents (those with sufficient length) in BoW format (besides its embeddings, in case the topic modeling algorithm in

use requires them). The complete procedure has been illustrated in Figure 3 for a concrete example.

The tool includes two different implementations that allow high parallelization of the processes just described. The first one is based on Spark, so the execution can be easily distributed among the nodes of a Spark cluster if such a cluster is available. If the latter condition cannot be met, the parallelization is achieved through Dask.

3.2 Topic modeling technologies

This section describes the topic modeling technologies included in the toolbox so far. All the algorithms are provided with default parameters as presented in their original works, but can be customized by the user for each specific training.

Mallet (McCallum, 2002) is a popular Java library with a highly efficient parallelizable implementation of LDA (Blei et al., 2003) based on collapsed Gibbs sampling known for providing good performance in terms of topic coherence and scalability. It puts the available resources to good use leading to fast training due to its multi-threading capabilities.

Spark-LDA Spark incorporates a machine learning library, MLLIB (Xiangrui Meng, et al., 2016) with two LDA implementations, both of them integrated into the ITMT: a fast online method based on Variational Bayes and another based on an expectation-maximization algorithm. Spark-LDA is suitable for fast training using horizontal scaling, but requires the use of a Spark cluster.

Neural Topic Models Bayesian-based topic models (BTMs) have been useful for text analysis for almost two decades, but neural topic models (NTMs) have gained research interest in the last years for their performance and flexibility. ITMT evaluates three representative NTM techniques, Autoencoded Variational Inference for Topic Models *(AVITM)-based implementation of LDA* and *ProdLDA* (Product-of-Experts LDA), both proposed in Srivastava and Sutton (2017); and Contextualized Topic Models (Bianchi et al., 2021a,b), against classical approaches for performance comparison.

Second-level hierarchical topic models Flat topic models like the ones presented above do not permit a topical analysis with the degree of granularity sometimes required by domain experts. In this line, we have included in the ITMT two novel implementations of hierarchical topic models (HTMs) that lack complicated implementations like most state-of-the-art HTMs, thus making it straightforward to integrate the knowledge of domain experts in the model building. Concretely, the integrated models are *HTM with word selection (HTM-WS)* and *HTM with document selection (HTM-DS)*, which follow a three-step process: 1) level-1 topic modeling and expansion topic selection; 2) level-2 model's synthetic training corpus construction by either keeping the words each level-1 corpus's document assigned to the topic selected for expansion (HTM-WS) or those documents with a proportion of the expansion topic larger than a customized threshold (HTM-DS); 3) training of this corpus to generated the level-2 model.

3.3 User-oriented tools

We present in this section the topic modeling user-oriented tools integrated into the ITMT.

Evaluation tools. The tools currently available are a set of global topics statistics and metrics:

- Topics' relative size in the corpus.
- Topics' chemical description with a penalty for the most common terms, instead of the traditional way of presenting the words in descending order of appearance frequency.
- Number of active documents (i.e., number of documents in which each topic is present), which helps distinguish between "vertical" and "horizontal" topics, i.e., topics that are specific to a limited number of documents vs topics that are shared among most documents.
- Entropy of the model, which gives an idea of whether a topic is characterized by a reduced number of terms or by a broad set (each of them in a smaller proportion).
- Coherence metrics, to provide insight into the degree of cohesion of the high probability terms for a given topic. It can be used as an indicator to help the user decide which topics are good candidates to be further split.

Visualization and annotation tools. pyLDAvis graphs (Sievert and Shirley, 2014) are generated for each trained model and embedded into the ITMT to ease the interpretation of the topics. To improve identification, the ITMT also supports the automatic labeling of topics and their posterior

modification through the user's manual labeling. For the automatic topic labeling system itself, i.e., the scheme that assigns to each collection of words characterizing one of the topics in the model a specific label from a customizable list of feasible terms, a zero-shot-classifier is used.

Curation tools. Aiming to improve the quality of the final model, the ITMT offers:

- Suggestions of similar topics, in the sense they co-occur with relative frequency, or the words characterizing them are the same.
- Fusion of topics (e.g., too similar topics) selected by the user, guaranteeing the probabilistic feasibility of the model.
- Sorting the topics of the model according to size, by placing the largest topics first.
- Topic deletion, which is useful to eliminate topics of little interest from the model.
- Topic model reset, allowing the user to discard all changes applied after training.

4 Existing frameworks

Probably the most widely used topic modeling frameworks are *Gensim* (Rehurek and Sojka, 2010) and Java-based package *Mallet* (McCallum, 2002), which include implementations of a handful of popular BTMs. They also provide pre-processing pipelines, hyper-parameters optimization, and the calculation of some coherence metrics. In the same direction, the *Stanford TMT* (Daniel Ramage, et al., 2009) is a set of topic modeling tools, including, inter alia, features such as the training of BTMs, the selection of parameters via a data-driven process, and the manipulation of texts from different spreadsheets.

Also released as a Java framework, *STTM* (Qiang et al., 2018) focuses on the integration of short text topic modeling algorithms, but it includes as well some long-text implementations and evaluation metrics. In Di Jiang, et al. (2021) the authors proposed a configurable framework named *Familia* that performs automatic parameter inference for a variety of topic models and supports the design of new topic models to best suit specific problems at hand. Aiming to bring additive regularization for topic modeling (ARTM) Vorontsov (2014); Kochedykov et al. (2017) accessible for the general public, Victor Bulatov, et al. (2020) proposed *TopicNet*, a Python module including a modular approach to topic model training and several visualization techniques, as well as semi-automated model selection and support for user-defined goal metrics.

Other state-of-the-art frameworks include *ToModAPI* (Lisena et al., 2020), a python-based API for the training, inference, and evaluation of different topic models; and *OCTIS* (Silvia Terragni, et al., 2021), also a Python-based framework + dashboard for the training of topic models, which additionally supports its analysis and comparison over several datasets and evaluation metrics, besides a bayesian-based hyperparameters optimization strategy. Lastly, BigML is a general tool for Machine Learning, that incorporates some Topic Modeling functionalities[4], including an optimized implementation of LDA for any text in seven languages, with preprocessing, training, inference, and visualization of models in a user-friendly dashboard, as well as the possibility of creating, configuring, and updating topic models programmatically via the BigML API and bindings.

Nonetheless, from the latter, only *ToModAPI* and *OCTIS* support the training of NTMs. Moreover, none of them allow the actual incorporation of knowledge expertise into the model building, nor allow for a thematic analysis with different levels of resolution. Hence, *ITMT* excels as an expert-in-the-loop oriented tool for the training at different resolution levels, curation, evaluation, and visualization of both BTMs and NTMs.

5 Conclusions

In this paper, we have presented IntelComp's Interactive Topic Model Trainer (ITMT), a Python-based tool that includes implementations of several state-of-the-art topic modeling algorithms orientated towards the usage of domain experts and a novelty implementation of second-level hierarchical topic models for granularity exploration. Moreover, the framework is provided with a set of tools for the evaluation, visualization, annotation, and curation of topic models, and a preprocessing pipeline.

For future work, we are active in offering each of the ITMT's components as a Docker container to transform the GUI into a web service.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innova-

[4] https://bigml.com/features/topic-model

tion program under grant agreement No 101004870, and from Grant TED2021-132366B-I00 funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR".

References

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021a. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proc. 59th Annual Meeting of the ACL and the 11th Intl. Joint Conf. on Natural Language Process. (Vol. 2: Short Papers)*, pages 759–766.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021b. Cross-lingual contextualized topic models with zero-shot learning. In *Proc. 16th Conf. EACL*, pages 1676–1683.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Daniel Ramage, et al. 2009. Topic modeling for the social sciences. In *NIPS 2009 Workshop on applications for topic models: text and beyond*, volume 5, pages 1–4.

Di Jiang, et al. 2021. Familia: A configurable topic modeling framework for industrial text engineering. In *Proc. Intl. Conf. Database Systems for Advanced Applications*, pages 516–528. Springer.

Denis Kochedykov, Murat Apishev, Lev Golitsyn, and Konstantin Vorontsov. 2017. Fast and modular regularized topic modelling. In *2017 21st Conf. Open Innovations Association (FRUCT)*, pages 182–193. IEEE.

Pasquale Lisena, Ismail Harrando, Oussama Kandakji, and Raphaël Troncy. 2020. Tomodapi: a topic modeling api to train, use and compare topic models. In *Proc. NLP-OSS*, pages 132–140.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. *http://mallet. cs. umass. edu*.

Jipeng Qiang, Yun Li, Yunhao Yuan, Wei Liu, and Xindong Wu. 2018. Sttm: A tool for short text topic modeling. *arXiv preprint arXiv:1808.02215*.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proc. LREC 2010 Workshop on new challenges for NLP frameworks*.

Carson Sievert and Kenneth Shirley. 2014. Ldavis: A method for visualizing and interpreting topics. In *Proc. Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 63–70.

Silvia Terragni, et al. 2021. OCTIS: comparing and optimizing topic models is simple! In *Proc. EACL: System Demonstrations*, pages 263–270.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

Victor Bulatov, et al. 2020. Topicnet: Making additive regularisation for topic modelling accessible. In *Proc. 12th LREC*, pages 6745–6752.

KV Vorontsov. 2014. Additive regularization for topic models of text collections. In *Doklady Mathematics*, volume 89, pages 301–304.

Xiangrui Meng, et al. 2016. Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1):1235–1241.