TOM: A library for topic modeling and browsing

Adrien Guille*, Edmundo-Pavel Soriano-Morales*

*Laboratoire ERIC, Université Lumière Lyon 2
adrien.guille@univ-lyon2.fr, edmundo.soriano-morales@univ-lyon2.fr

Abstract. In this paper, we present TOM (TOpic Modeling), a Python library for topic modeling and browsing. Its objective is to allow for an efficient analysis of a text corpus from start to finish, via the discovery of latent topics. To this end, TOM features advanced functions for preparing and vectorizing a text corpus. It also offers a unified interface for two topic models (namely LDA using either variational inference or Gibbs sampling, and NMF using alternating least-square with a projected gradient method), and implements three state-of-the-art methods for estimating the optimal number of topics to model a corpus. What is more, TOM constructs an interactive Web-based browser that makes exploring a topic model and the related corpus easy.

1 Introduction

Topic models are useful tools for unveiling the latent topical structure of text corpora. They can make searching, browsing and summarizing these corpora easier. Several models and algorithms to approximate them have been proposed in the recent years. The quality of the discovered topics depends on the model, the approximation algorithm, the nature of the corpus being studied, as well as the number of topics (Stevens et al., 2012). Therefore, in order to perform an efficient topic-based analysis of a text corpus, it is important to compare several approaches to identify the most relevant topics. However, this is a difficult task because the existing implementations of the approximation algorithms are independent, which means that one has to learn how data are structured in each implementation; what the functions to manipulate each topic model are; how to fit these topic models on the exact same set of features, *etc*. On the other hand, several methods have been proposed to estimate the optimal number of topics to model a corpus, but – to the best of our knowledge – their implementations aren't publicly available.

In this short paper we present TOM (TOpic Modeling), an open source library written in Python for analyzing a text corpus from start to finish, via the discovery of latent topics. Apart from advanced corpus preparation functions, TOM offers a unified interface for existing robust implementations of approximation algorithms, that makes fitting and manipulating topic models easy. It also implements several functions to estimate the optimal number of topics to model a corpus. What is more, TOM can automatically build a Web based interface for exploring a topic model and a corpus in an interactive manner.

2 Proposed library

In this section, we first describe the capabilities of the proposed library, TOM, then we illustrate how to use it with the help of short code snippets. Sources and documentation are available online at *https://github.com/AdrienGuille/TOM*.

2.1 Features

TOM operates on a text corpus, optionally supplemented with meta-data such as authors or dates of writing/publication.

Corpus preparation Preparing the corpus is fundamental, in the sense that the relevancy of all further processing depends on this step. Advanced preparation functions are available for both French and English. TOM can lemmatize French using MElt, a maximum entropy Markov model-based part-of-speech tagging system especially designed for French (Denis and Sagot, 2012), and Lefff, a morphological and syntactic lexicon for French (Sagot, 2010), to match $\{$word, part-of-speech$\}$ pairs with lemmas. It can also lemmatize English in a similar manner, using another maximum entropy model trained for English (Bird et al., 2009) and the WordNet lexicon (Miller, 1995). Eventually, TOM constructs the vector space representation with unigrams or n-grams as features, using either $tf \cdot idf$ or simply tf. The vector space is a $n \times m$ matrix, with n the number of texts and m the number of features.

Topic models Given the vector space representation of a corpus and a small number of topics k ($k \ll m$), a topic model consists of two matrices: W and H. W is a $n \times k$ matrix that describes the texts in terms of topics, and H is a $k \times m$ matrix that describes topics in terms of features (*i.e.* words or n-gram of words). More precisely, the coefficient $w_{i,j}$ defines the importance of topic j in text i, and the coefficient $h_{i,j}$ defines the importance of feature j in topic i. Two topic models are available in TOM: (i) Latent Dirichlet Allocation (LDA), a probabilistic generative topic model proposed by Blei et al. (2003), and (ii) Non-negative Matrix Factorization (NMF), a vector space factorization method which has recently became popular for topic modeling (Berry and Browne, 2005). Concerning LDA, the library provides two approximation algorithms: the original variational inference algorithm and the Gibbs sampling variant (Griffiths, 2004). Concerning NMF, it relies on an algorithm based on alternating least-square with projected gradient descent (Lin, 2007).

Parameter estimation Choosing an appropriate number of topics is critical to ensure a pertinent modeling of a text corpus. TOM implements three methods to guide this choice: (i) the stability-based method proposed by Greene et al. (2014), (ii) the consensus-based method proposed by Brunet et al. (2004), and (iii) the divergence-based method proposed by Arun et al. (2010). Each of these methods is based on a particular assumption and leads to the computation of a specific metric, which value is related to the quality of a topic model for a corpus and a given number of topics. TOM also offers functions to plot these metrics in order to facilitate their visual inspection.

Topic model browser The topic model browser offers 3 overviews: the author index, the complete vocabulary and the topic cloud, where each topic is represented by a bubble labeled with the most relevant words and which diameter is proportional to its overall frequency. It also offers interactive detailed views for: each topic, each document, each author, and each feature (*i.e.* words or n-gram of words) of the vector space. For instance, the detailed view about a topic presents the most relevant features, the evolution of the topic frequency through time, the list of related texts and the collaboration network that links authors. The detailed view for a text presents the most significant features, the topic distribution and the most similar texts. Note that some elements may be missing, depending on the meta-data available with the input corpus.

2.2 Usage

Load and prepare a text corpus The following code snippet shows how to load a corpus of French documents, lemmatize them and vectorize them using unigrams and $tf \cdot idf$.

```
corpus = Corpus(source_file_path='input/raw_corpus.csv',
                language='french', # language for stop-words
                vectorization='tfidf',
                n_gram=1,
                max_relative_frequency=0.8,
                min_absolute_frequency=4,
                preprocessor=FrenchLemmatizer())
print 'corpus size:', corpus.size
print 'vocabulary size:', len(corpus.vocabulary)
print 'Vector for document 0:\n', corpus.vector_for_document(0)
```

Instantiate a topic model and estimate the optimal number of topics Here, we instantiate a NMF based topic model and generate plots for the three metrics to estimate the optimal number of topics to model the loaded corpus.

```
topic_model = NonNegativeMatrixFactorization(corpus)
viz = Visualization(topic_model)
viz.plot_greene_metric(min_num_topics=5, max_num_topics=50,
                       tao=10, step=1, top_n_words=10)
viz.plot_arun_metric(min_num_topics=5, max_num_topics=50,
                     iterations=10)
viz.plot_consens_metric(min_num_topics=5, max_num_topics=50,
                        iterations=10)
```

To use LDA with Gibbs sampling instead of NMF, one only has to replace the first line with the following:

```
topic_model = LatentDirichletAllocation(corpus, method='gibbs')
```

Infer a topic model and save/load it To allow reusing previously learned topics models, TOM can save them on disk, as shown below.

```
topic_model.infer_topics(num_topics=15)
utils.save_topic_model(topic_model, 'output/NMF_15topics.tom')
topic_model = utils.load_topic_model('output/NMF_15topics.tom')
```

Print information about a topic model This code excerpt illustrates how one can manipulate a topic model, *e.g.* get the topic distribution for a text (a vector of length k) or the word distribution for a topic (a vector of length m).

```
print '\nTopic distribution for document 0:', \
    topic_model.topic_distribution_for_document(0)
print '\nWord distribution for topic 0:', \
    topic_model.word_distribution_for_topic(0)
```

To facilitate interactions with topic models, TOM offers higher level functions, *e.g.* to identify the topic with the highest weight for a given document, to get the most relevant words for a given topic, or quickly print all the topics.

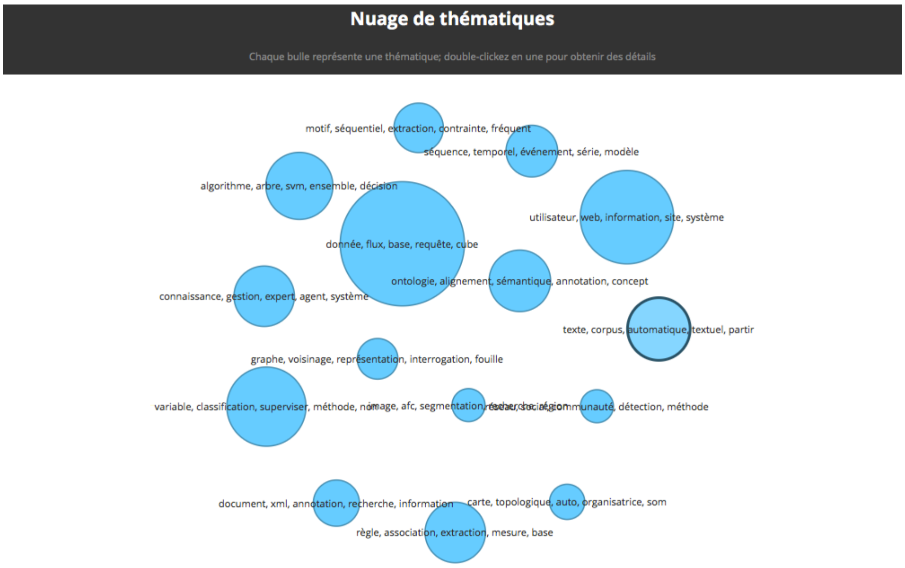
```
print '\nMost likely topic for document 0:', \
    topic_model.most_likely_topic_for_document(0)
print '\n10 most relevant words for topic 0:', \
    topic_model.top_words(0, 10)
print '\nTopics:'
topic_model.print_topics(num_words=10)
```

3 Demonstration

In this demo, the audience will be invited to explore the EGC anthology (817 articles published from 2004 until 2015) through a topic model browser automatically constructed by TOM. See Guille et al. (2016) for more details about this topic model. Fig. 1 and Fig. 2 (page 5) give a glimpse of what the participants will have access to. They respectively show the topic cloud overview of the anthology, and details about one of the topics. This topic model browser can be accessed online at *http://mediamining.univ-lyon2.fr/people/guille/egc2016*. The attendees will also have the opportunity to explore another topic model browser based on the transcripts of 900+ speeches made by François Hollande as the President of France, between May 2012 and January 2016.

4 Future work

Future work includes adding more topic models and approximation algorithms. Most notably, it would be interesting to implement an approximation algorithm that optimizes the Kullback–Leibler divergence objective function, since Ding et al. (2008) have shown that NMF is then equivalent to Probabilistic Latent Semantic Analysis (PLSA), a seminal topic model proposed by Hofmann (1999).

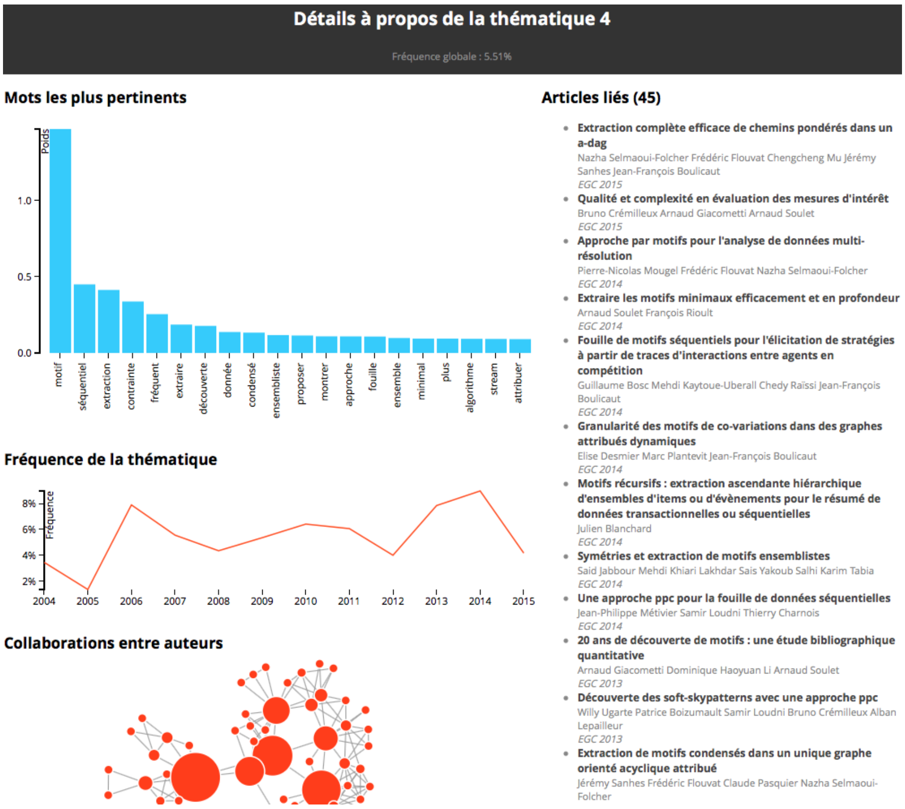


Nuage de thématiques

Chaque bulle représente une thématique; double-clickez en une pour obtenir des détails

motif, séquentiel, extraction, contrainte, fréquent
séquence, temporel, événement, série, modèle
algorithme, arbre, svm, ensemble, décision
utilisateur, web, information, site, système
donnée, flux, base, requête, cube
ontologie, alignement, sémantique, annotation, concept
connaissance, gestion, expert, agent, système
texte, corpus, automatique, textuel, partir
graphe, voisinage, représentation, interrogation, fouille
variable, classification, superviser, méthode, nom
image, afc, segmentation, recherche, région
réseau, social, communauté, détection, méthode
document, xml, annotation, recherche, information
carte, topologique, auto, organisatrice, som
règle, association, extraction, mesure, base

FIG. 1 – *Overview of the corpus with the topic cloud.*



Détails à propos de la thématique 4

Fréquence globale : 5.51%

Mots les plus pertinents

Fréquence de la thématique

Collaborations entre auteurs

Articles liés (45)

- **Extraction complète efficace de chemins pondérés dans un a-dag**
  Nazha Selmaoui-Folcher Frédéric Flouvat Chengcheng Mu Jérémy Sanhes Jean-François Boulicaut
  EGC 2015
- **Qualité et complexité en évaluation des mesures d'intérêt**
  Bruno Crémilleux Arnaud Giacometti Arnaud Soulet
  EGC 2015
- **Approche par motifs pour l'analyse de données multi-résolution**
  Pierre-Nicolas Mougel Frédéric Flouvat Nazha Selmaoui-Folcher
  EGC 2014
- **Extraire les motifs minimaux efficacement et en profondeur**
  Arnaud Soulet François Rioult
  EGC 2014
- **Fouille de motifs séquentiels pour l'élicitation de stratégies à partir de traces d'interactions entre agents en compétition**
  Guillaume Bosc Mehdi Kaytoue-Uberall Chedy Raïssi Jean-François Boulicaut
  EGC 2014
- **Granularité des motifs de co-variations dans des graphes attribués dynamiques**
  Elise Desmier Marc Plantevit Jean-François Boulicaut
  EGC 2014
- **Motifs récursifs : extraction ascendante hiérarchique d'ensembles d'items ou d'évènements pour le résumé de données transactionnelles ou séquentielles**
  Julien Blanchard
  EGC 2014
- **Symétries et extraction de motifs ensemblistes**
  Said Jabbour Mehdi Khiari Lakhdar Sais Yakoub Salhi Karim Tabia
  EGC 2014
- **Une approche ppc pour la fouille de données séquentielles**
  Jean-Philippe Métivier Samir Loudni Thierry Charnois
  EGC 2014
- **20 ans de découverte de motifs : une étude bibliographique quantitative**
  Arnaud Giacometti Dominique Haoyuan Li Arnaud Soulet
  EGC 2013
- **Découverte des soft-skypatterns avec une approche ppc**
  Willy Ugarte Patrice Boizumault Samir Loudni Bruno Crémilleux Alban Lepailleur
  EGC 2013
- **Extraction de motifs condensés dans un unique graphe orienté acyclique attribué**
  Jérémy Sanhes Frédéric Flouvat Claude Pasquier Nazha Selmaoui-Folcher

FIG. 2 – *Detailed view about one of the discovered topics.*

References

Arun, R., V. Suresh, C. V. Madhavan, and M. N. Murthy (2010). On finding the natural number of topics with latent dirichlet allocation: Some observations. In *PAKDD*, pp. 391–402.

Berry, M. W. and M. Browne (2005). Email surveillance using non-negative matrix factorization. *Journal of Computational and Mathematical Organization Theory 11*(3), 249–264.

Bird, S., E. Klein, and E. Loper (2009). *Natural Language Processing with Python*. O'Reilly Media.

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

Brunet, J., P. Tamayo, and J. Golub, T.R.and Mesirov (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences 101*(12), 4164—4169.

Denis, P. and B. Sagot (2012). Coupling an annotated corpus and a lexicon for state-of-the-art pos tagging. *Language Resources and Evaluation 46*(4), 721–736.

Ding, C., T. Lib, and W. Peng (2008). *Computational Statistics and Data Analysis* (52), 3913––3927.

Greene, D., D. O'Callaghan, and P. Cunningham (2014). How many topics? stability analysis for topic models. In *ECML PKDD*, pp. 498–513.

Griffiths, Thomas L., S. M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences 101*, 5228–5235.

Guille, A., E. P. Soriano Morales, and C. O. Truica (2016). Topic modeling and hypergraph mining to analyze the EGC conference history. In *EGC*, pp. 383–394.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *SIGIR*, pp. 50–57.

Lin, C. J. (2007). Projected gradient methods for non-negative matrix factorization. *Neural Computation 19*, 2756–2779.

Miller, G. A. (1995). Wordnet: A lexical database for english.

Sagot, B. (2010). The lefff, a freely available and large-coverage morphological and syntactic lexicon for french. In *LREC*.

Stevens, K., P. Kegelmeyer, D. Andrzejewski, and D. Buttler (2012). Exploring topic coherence over many models and many topics. In *EMNLP-CoNLL*, pp. 952–961.

Résumé

Cet article présente TOM, une bibliothèque Python pour la modélisation et l'exploration de thématiques dont l'objectif est de permettre de mener une analyse efficace, de bout en bout, d'un corpus textuel via la découverte de thématiques latentes. TOM offre des fonctions pour la préparation et la vectorisation de corpus, une interface unifiée pour deux modèles de thématiques (LDA et NMF), et implémente trois méthodes pour estimer le nombre optimal de thématiques. Par ailleurs, TOM construit automatiquement un explorateur interactif permettant facilement d'étudier un modèle de thématiques et les documents liés.