Multi-Word Structural Topic Modelling of ToR Drug Marketplaces

Stefano Guarino and Mario Santoro
Istituto per le Applicazioni del Calcolo “Mauro Picone”
Consiglio Nazionale delle Ricerche
Via dei Taurini, 19, Rome, Italy
Email: {s.guarino,m.santoro}@iac.cnr.it

Abstract

Topic Modelling (TM) is a widely adopted generative model used to infer the thematic organization of text corpora. When document-level covariate information is available, so-called Structural Topic Modelling (STM) is the state-of-the-art approach to embed this information in the topic mining algorithm. Usually, TM algorithms rely on unigrams as the basic text generation unit, whereas the quality and intelligibility of the identified topics would significantly benefit from the detection and usage of topical phrasemes*. Following on from previous research, in this paper we propose the first iterative algorithm to extend STM with n-grams, and we test our solution on textual data collected from four well-known ToR drug marketplaces. Significantly, we employ a STM-guided n-gram selection process, so that topic-specific phrasemes can be identified regardless of their global relevance in the corpus. Our experiments show that enriching the dictionary with selected n-grams improves the usability of STM, allowing the discovery of key information hidden in an apparently “mono-thematic” dataset.*

Keywords

STM, N-grams, Tor, Markets

1. Introduction and Background

Topic Modelling (TM) is a widely adopted text-mining technique that allows understanding the underlying semantic structure of a corpus of documents. From a purely quantitative point of view, documents may be viewed as joint memberships of text *tokens* (words, phrases, paragraphs, ...) which have various patterns of (co-)occurrence. TM is based on explaining these patterns through a *generative* random process in which corpus-wide *latent variables*, referred to as *topics*, are combined according to document-specific *mixtures*. Each topic is characterized by a unique probability distribution over tokens, and each document is assumed to have been assembled token by token by iteratively drawing a topic from the document’s random mixture and a token from the selected topic. The documents are the only observable data values of this process and are used by TM algorithms for inferring the topics and the mixtures [1] in order to discover the thematic organization of the corpus and the composition of each document.

A significant body of work deals with finding the optimal definition of text token. Traditionally, TM follows the diffused *Bag Of Words* (BOW) paradigm that represents documents as multisets of words, ignoring grammar and syntax to only focus on *unigrams* frequencies. However, BOW-based TM algorithms suffer from two limitations: inability to detect topical multi-word expressions (*i.e.*, *phrasemes*), and difficulty in visualizing/interpreting the obtained topics. To make up for it, two main directions have been explored in the literature: extending the generative model to include n-gram tokens (TNG [2], PD-LDA [3]), or running standard TM and identifying topical phrasemes as a post-process (Turbo Topics [4], KERT [5]). ToPMine [6] follows a third approach: it identifies the most suitable text segmentation during a pre-processing stage and uses the obtained *bag of phrases* as a constraint in the topical inference process. As a result, ToPMine is more scalable and produces top-quality topical n-grams with respect to previous work, improving the usability of the obtained topics.

ToPMine enjoys the pros of expanding text tokens *before* TM, without the drawbacks of embedding n-grams in the model. However, ToPMine’s phrase mining algorithm may not be optimal for all corpora. The algorithm is based on comparing absolute/relative phrase frequency estimators of words and word sequences, and phrases that are key to a *single* topic but composed of frequent and/or not topic/context-specific words may not match the *collocation* requirement and be discarded. Additionally, ToPMine is a complete framework which embeds a specifically designed variant of TM called PhraseLDA, whereas standard TM algorithms are already available in a number of scientific R/Python/Java libraries. It is therefore interesting investigating the ability of existing TM implementations to work with variable length tokens, automatically filtering phrasemes based on their topical relevance. A TM-driven phrase selection algorithm, although not scalable to very large corpora, would better suit the characteristics of datasets that present peculiar idioms and require a fine-grained analysis.

To test the viability of a similar solution, in this paper we present an iterative algorithm to identify topical *phrasemes* and to produce more informative topics, evaluating its performance on the textual data collected from four important ToR drug marketplaces. Specifically, we focus on a recent extension to TM called *Structural Topic Modelling* [7] (STM), which allows incorporating tags, categories, metadata and other instrumental information accompanying the text archive. STM uses this *covariate* information to parametrize the prior distributions in such a way to potentially affect both topical

0-7695-6360-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICSC.2018.00048

prevalence and topical content. To the best of our knowledge, ours is the first ever attempt at defining an STM-guided n-gram selection process, thus representing a preliminary study of both the impact of n-grams on the quality of STM and the ability of STM to detect topical n-grams regardless of their global relevance in the corpus. The proposed algorithm refines the characterization of the dataset, identifying intelligible topics and letting unexpected/hidden information emerge quite naturally.

Studying the semantics of ToR pages is a viable approach to gain insights into its structure and vulnerabilities [8]. Specific ToR-oriented web mining toolkits have been proposed [9], and TM has already been used to classify text obtained from over a thousand ToR hidden services [10]. Previous work showed that resources devoted to criminal activities constitute the core of ToR content [11], but the literature lacks a more detailed analysis of the thematic organization of ToR marketplaces. By focusing on a specific subset of the ToR network for which partial classification is available beforehand, we show that advanced topic modelling techniques can be used to fully characterize an apparently "mono-thematic" corpus, providing a deeper understanding of drug offer and demand over ToR.

The remainder of this paper is organized as follows: in Section 2 we describe our extended STM algorithm; in Section 3 we report the main results of our analysis; finally, in Section 4 we present concluding remark and suggest viable directions for future work.

2. Our Extended STM

Within the scope of the "IANCIS" ISEC Project[1], we designed a ToR-oriented web mining toolkit which includes ad-hoc spiders for focused and semi-automated web scraping [9]. The ability of these spiders to deal with login procedures or captcha solvers allowed performing a complete crawling of four famous ToR drug marketplaces: Alphabay, Crypto Market, East India and Nucleus. Since these marketplaces are mostly composed of pages advertising some item for sale and grouped by category, extracting all text from this dataset yields a corpus of documents for which *covariate* information (the market and the category) are immediately available.

Despite the characteristics of this dataset make STM a natural choice, the presence of context-specific idioms and, more generally, a *topical slang* suggests the inadequacy of any algorithm relying on a strict application of the Bag Of Words (BOG) model. We therefore designed an iterative algorithm based on the rationale that adding an idiom to the dictionary helps topics extraction and characterization only as long as the idiom and its components express different concepts that are relevant to different topics. In practice, standard STM without covariates modeling is iteratively used to detect topic-relevant token-pairs which are merged into a single *extended-word*, up to a moment when no more relevant compound terms emerge. As a result, the obtained topics are better characterized and more intelligible.

2.1. The Algorithm

Let $D = \{d_1, \ldots, d_m\}$ be our corpus. At the very beginning, each document d_j is just a string of characters. Step zero of our algorithm therefore consists in preprocessing the documents as follows: (i) removing special characters and forcing the string to lowercase; (ii) tokenizing the string into words; (iii) removing stop-words (*i.e.*, function words, auxiliary/lexical verbs, adverbs/adjectives, file extensions, ...). As a result, each document is formatted as an ordered list of clean/filtered words that we denote $d_j = (w_j^1, \ldots, w_j^{n_j})$.

The proposed iterative solution is reported in Algorithm 1, where, to improve readability, we use the following notation: given two *tokens* w_1 and w_2, $w_1\_w_2$ denotes their concatenation, where w_1 and/or w_2 may in turn be the concatenation of any number of words; $f(d_j, w)$ denotes the tf-idf of (compound) word w in document d_j; if K is the number of topics, $|W|$ the total number of tokens and $|D|$ the number of documents, F is the $|W| \times K$ matrix whose entry $F_{l,t}$ is the FREX[2] score of word (or token) w_l with respect to topic t, whereas P is the $K \times |D|$ matrix whose entry $P_{t,j}$ is the probability of topic t appearing in document d_j; finally, the product $S = F \cdot P$ yields a $|W| \times |D|$ matrix whose entry $S_{l,j}$ measures the relevance of word w_l in document d_j. In words, the algorithm iteratively *extends* the corpus dictionary by adding all tokens $w_j^i\_w_j^{i+1}$ obtained concatenating any two consecutive words/tokens in a document. After a simple STM without covariates is implemented, for each document d_j only compound tokens whose score $S_{l,j}$ is above a predetermined thresholds s_{min} are kept, and every occurrence of the corresponding pair in d_j is replaced by the unique extended token. The algorithm stops when at least a fraction ϵ of new relevant tokens are found in total, where ϵ is a parameter. Only the final topic modelling is considered.

3. Results

In this section we present the main results of the application of our Extended STM to text data collected from four well-known ToR marketplaces. In total, the dataset consists of 20491 html pages used to advertise and sell several drugs-related goods, divided as follows:

Nucleus	: 8902 pages
Alphabay	: 7472 pages
Crypto Market	: 2435 pages
East India	: 1682 pages

As reported in Section 2, we also collected the category tags that each marketplace uses to label its pages. In an attempt to guarantee cross-market uniformity, we aggregated a few clearly analogous categories which were differently tagged in

[1] DG Home Affairs ISEC Programme 2013 - Project IANCIS "Indexing of Anonimous Networks for Crime Information Search", 2014-2016, GA n. HOME/2013/ISEC/AG/INT/4000005222 - www.iancis.eu

[2] FREX: FRequency And EXclusivity score. See [12]

Algorithm 1:

```
Data: parameters $K \in \mathbf{N}$, $\epsilon \in (0,1)$, $s_{min} \in (0,1)$
repeat
    foreach $j = 1, \ldots, m$ do
        initialize $W_j \leftarrow \cup_{w \in d_j}\{w\}$ and $W'_j \leftarrow \emptyset$;
        foreach $w_j^i, w_j^{i+1} \in d_j$ do
            update $W'_j \leftarrow \{w_j^i\_w_j^{i+1}\}$;
        end
    end
    initialize $W \leftarrow \cup_{j=1}^m (W_j \cup W'_j)$ and $W' \leftarrow \cup_{j=1}^m W'_j$;
    foreach $j = 1, \ldots, m$ and $w_k \in W$ do
        inizialize $X(j,k) \leftarrow f(d_j, w_k)$;
    end
    run STM on $X$ with $K$ topics and store FREX scores in matrix $F$ and
      topic distributions in matrix $P$;
    compute $S = F \cdot P$;
    initialize $R \leftarrow \emptyset$;
    foreach $t = 1, \ldots, K$ do
        initialize $R_t \leftarrow \emptyset$;
        foreach (compound) $w_l \in W'$ do
            if $S_{l,j} > s_{min}$ then
                update $R_t \leftarrow R_t \cup \{w_l\}$;
            end
        end
        update $R \leftarrow R \cup R_t$;
    end
    foreach (compound) $w_l \in R$ do
        foreach $w_j^i, w_j^{i+1} \in d_j$ do
            if $w_l = w_j^i\_w_j^{i+1}$ then
                $d_j \leftarrow (\ldots, w_j^{i-1}, w_l, w_j^{i+2}, \ldots)$;
            end
        end
    end
until $|R| > \epsilon|W|$;
```

different markets[3], obtaining 39 possible categories from the initial set of 44 collected tags. In Table 1 we report the per-market distribution of the overall top-ten categories.

TABLE 1. Per-market frequencies of the top-ten categories.

Category	Frequencies			
	Nucleus	Alphabay	Crypto	East India
cannabisandhashish	0.2641	0.1878	0.2920	0.3210
ecstasy	0.2407	0.0278	0.0312	0.1617
stimulants	0.1495	0.2655	0.0230	0.1718
prescription	0.0722	0.1171	0.0464	0.0375
opioids	0.0614	0.1178	0.0181	0.0523
psychedelics	0.0523	0.1560	0.0136	0.1046
benzodiazepines	0.0803	0.0000	0.0201	0.0559
steroids	0.0450	0.0683	0.0230	0.0345
dissociatives	0.0274	0.0001	0.0033	0.0262
cocaine	0.0000	0.0000	0.0973	0.0000

3.1. STM detection of topic-relevant token-$n$plets

As described in section 2 we runned an iterative STM without covariates modelling for token-$n$plets with $n \in (1,2,3)$ in every run. We setted, using empirical considerations, $p_{min} = 0.01$, $frex_{min} = 0.95$ and $\epsilon = 10^{-7}$. The latter, multiplied by $|W|$, is about 100 tokens.

We want to emphasize that at this step we are only interested in topic-relevant token-$n$plets and not to find an extremely good number of topics K. So we used the *spectral initialization* [13], an automatic procedure of STM that provide an useful starting point for K.

3.2. STM Tuning: Selecting K (Number of Topics)

Once defined the relevant token-$n$plets, is time to run the final STM using a covariates modeling.

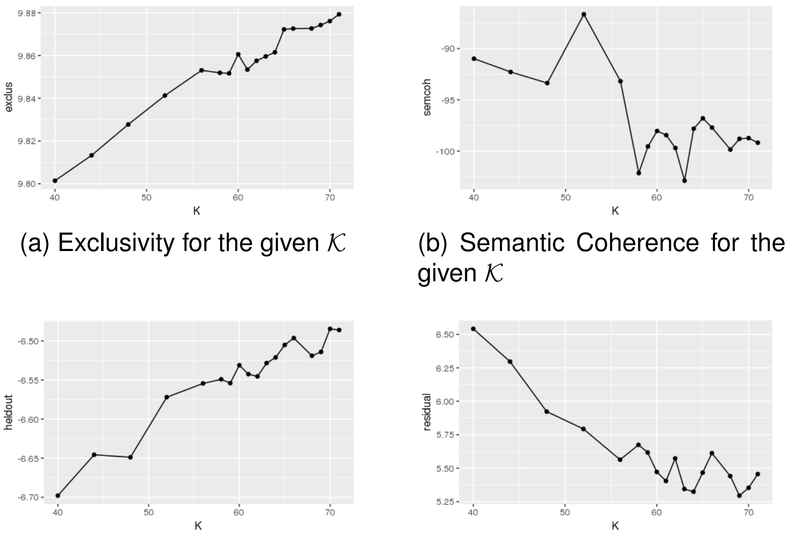
First of all we needed to find a good value for the parameter K (the number of topics). Although K has a significant impact on the outcomes of any Topic Modelling implementation, there is no established methodology in the literature to find the "correct" value for K for a given corpus [14]. As a remedy, STM [7] comes with a *searchK* function that allows comparing candidate values for K on the basis of 4 different data-driven tests: exclusivity, semantic coherence, heldout, and residual. To select the most suitable K we used *searchK* on the set $\mathcal{K} = \{40, 44, 48, 52, 56, 58, 59, 60, 61, 62, 63, 64, 65, 66, 68, 69, 70, 71\}$. The rationale for our choice of $\mathcal{K}$ was to explore a relatively wide range of values larger than 39 (the number of categories), in order to assess the ability of Topic Modelling to (automatically) produce a refined characterization of the dataset and to extract cross-category topics such as "shipment" or "drugs effects". Figure 1 reports the results of running *searchK* on $\mathcal{K}$. It can be seen that exclusivity and heldout roughly increase with K, whereas semantic coherence and residual behave almost contrarily, reflecting the idea that the former measure the average specificity of a topic and the latter its thoroughness. We decided to set $K = 65$ as it seems to provide a reasonable trade-off among the four metrics.

3.3. A zoom to the topic 30: Methamphetamine

Using the extended-STM, we found very easily that topic number 30 was better describing pages selling methamphetamine. In details the highest score tokens were: ice, meth, `crystal_meth`, shards, `crystal_methamphetamine`, `0.5g_crystal_methamphetamin`. It is clearly that the use of n-grams increase the specificity, the readability and the understanding of the topic.

Respect to the given per-market categories, methamphetamine was tagged only in Crypto Market with frequency of 0.022. Now, using the estimateEffect we can see if there is a statistically significant difference for the prevalence of this topic. As a result, respect to Alphabay, in Nucleus the methamphetamine topic is about 2 times more prevalent, from 0.0114 to 0.0218.

[3] Specifically, we aggregated as follows: *cannabisandhashish = cannabisandhashish + cannabis + hash + weed*; *benzodiazepines = benzodiazepines + benzos*; *opioids = opioids + opium*.



(a) Exclusivity for the given $\mathcal{K}$

(b) Semantic Coherence for the given $\mathcal{K}$

(c) Heldout for the given $\mathcal{K}$

(d) Residual for the given $\mathcal{K}$

Fig. 1. Plots of the values of the four tests for every $K \in \mathcal{K}$

In order to compare our results with those from TopMine[4] we verified that, for the most representative document of the topic, TopMine is able to find the same N-grams except `0.5g_crystal_methamphetamin`.

3.4. A zoom to the cannabis and hashish topics

In the same way as the previous section, we found 7 different topics: $\{1, 14, 22, 46, 50, 54, 56\}$. Focusing attention to topic 56, once again points out the improve of the extended-STM. The highest score tokens were: `shatter_pull_snap`, `sour_strawberry_diesel`, `og_kush`,`ak_strain`, indoor, scout, hybrid, indica, sativa, `chemicalscannabis_hashishbuds`, `content_thc_cbd`, `14g_black_diamond`. The effect of the Market covariate show that there is a 30% increase in prevalence in East India respect to Alphabay. Like the previuos subsection, for topic 22, we verified that TopMine is able to find the same N-grams except `14g_black_diamond`, `shatter_pull_snap`, `ak_strain`.

4. Conclusions

In this paper, we presented an iterative algorithm for extending Structural Topic Modelling (STM) with topical n-grams, and we tested our algorithm analyzing the corpus of text documents obtained crawling the pages of four well-known drug marketplaces of the ToR web. The availability of document-level covariates makes STM a natural choice for similar datasets, but any algorithm relying on a strict application of the Bag Of Words (BOW) language model struggles in the presence of a *topical slang* where context-specific idioms frequently occur. As an exploratory approach we therefore opted for an ad-hoc heuristic based on iteratively apply standard STM to detect topic-relevant term-pairs and merge them into a single *extended-word*. After a few rounds no more relevant compound terms turned up, whereas the coherence and the intelligibility of the obtained topics were significantly enhanced. As a result, through a fine-grained and cross-market analysis of the thematic organization of the corpus we were able to gain relevant information about drug trade on ToR that goes well beyond those provided by the already available high level content classification.

References

[1] D. M. Blei, “Probabilistic topic models,” *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[2] X. Wang, A. McCallum, and X. Wei, “Topical n-grams: Phrase and topic discovery, with an application to information retrieval,” in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 697–702.

[3] R. V. Lindsey, W. P. Headden III, and M. J. Stipicevic, “A phrase-discovering topic model using hierarchical pitman-yor processes,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 214–222.

[4] D. M. Blei and J. D. Lafferty, “Visualizing topics with multi-word expressions,” *arXiv preprint arXiv:0907.1013*, 2009.

[5] M. Danilevsky, C. Wang, N. Desai, X. Ren, J. Guo, and J. Han, “Automatic construction and ranking of topical keyphrases on collections of short documents,” in *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 2014, pp. 398–406.

[6] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han, “Scalable topical phrase mining from text corpora,” *Proceedings of the VLDB Endowment*, vol. 8, no. 3, pp. 305–316, 2014.

[7] M. E. Roberts, B. M. Stewart, D. Tingley, E. M. Airoldi *et al.*, “The structural topic model and applied social science,” in *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*, 2013.

[4] We used the source code from authors at http://illimine.cs.uiuc.edu/software/topmine/

[8] M. Bernaschi, A. Celestini, S. Guarino, and F. Lombardi, “Exploring and analyzing the tor hidden services graph,” *ACM Transactions on the Web (TWEB)*, vol. 11, no. 4, p. 24, 2017.

[9] A. Celestini and S. Guarino, “Design, implementation and test of a flexible tor-oriented web mining toolkit,” in *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics*. ACM, 2017, p. 19.

[10] M. Spitters, S. Verbruggen, and M. van Staalduinen, “Towards a comprehensive insight into the thematic organization of the tor hidden services,” in *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*, Sept 2014, pp. 220–223.

[11] A. Biryukov, I. Pustogarov, F. Thill, and R.-P. Weinmann, “Content and popularity analysis of tor hidden services,” in *Distributed Computing Systems Workshops (ICDCSW), 2014 IEEE 34th International Conference on*. IEEE, 2014, pp. 188–193.

[12] J. M. Bischof and E. M. Airoldi, “Summarizing topical content with word frequency and exclusivity,” in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 201– 208.

[13] M. Roberts, B. Stewart, and D. Tingley, “Navigating the local modes of big data: The case of topic models,” in *Computational Social Science: Discovery and Prediction*. Cambridge University Press, 2016, pp. 51 – 97.

[14] J. Grimmer and B. M. Stewart, “Text as data: The promise and pitfalls of automatic content analysis methods for political texts,” *Political analysis*, vol. 21, no. 3, pp. 267–297, 2013.