Topical Phrase Extraction from Clinical Reports by Incorporating both Local and Global Context

Gabriele Pergola, Yulan He and **David Lowe**
School of Engineering and Applied Science, Aston University, UK
pergolag@aston.ac.uk, y.he@cantab.net, d.lowe@aston.ac.uk

Abstract

Making sense of words often requires to simultaneously examine the surrounding context of a term as well as the global themes characterizing the overall corpus. Several topic models have already exploited word embeddings to recognize local context, however, it has been weakly combined with the global context during the topic inference. This paper proposes to extract topical phrases corroborating the word embedding information with the global context detected by Latent Semantic Analysis, and then combine them by means of the Pólya urn model. To highlight the effectiveness of this combined approach the model was assessed analyzing clinical reports, a challenging scenario characterized by technical jargon and a limited word statistics available. Results show it outperforms the state-of-the-art approaches in terms of both topic coherence and computational cost.

Introduction

Topic models have been extensively used to generate synthetic representations of the main themes characterizing a large document collection. Documents are traditionally represented under the bag-of-words assumption, a simple but effective representation that ignores the word orders, but in spite of this has shown worth noting results (**?**). However, this assumption has commonly led to the extraction of unigram topics relying on the word co-occurrence patterns across documents. This has notably narrowed the topic expressiveness as the shared semantic of words is solely based on the global context; moreover, many domain-specific documents might include concepts that are unfolded in multiple words rather than in a single term. Clinical reports are a prominent example of this family as medical concepts are often expressed in terms of multi-word phrases. For example, the phrases "*white blood cell*" or "*blood sugar*" would lose their meaning if decomposed as unigrams; in addition, the word *cell* and *sugar* might be wrongly put under the same topic because of the shared *blood* term.

Recently, word embeddings have gained an increasing interest thanks to their improved efficiency in representing words as continuous vectors of a low-dimensional space (**?**; **?**). The resulting embeddings have been proved to encode numerous semantic relations (e.g. similarity or analogies) based on the local context of words (**?**). Several works have combined topic models with word embedding (**?**; **?**). This has enhanced the semantic coherence of the topics discovered since words with similar semantic and syntactic properties are close to each other in the embedding space which overcomes the curse of dimensionality when representing words as atomic units. However, these models commonly entail a significant computational cost and are exposed to the *topic shifting* problem (**?**). Indeed, words that share similar context windows might potentially be treated as directly co-related into the embedding space, misleading word similarity in case of antonyms (e.g. *tall* and *short*) or co-hyponyms (e.g. *schizophrenia* and *alzheimer*).

The computational cost required to combine word embeddings and topic models can be reduced by adopting the *Generalized Pólya urn model* (**?**). Although the Latent Dirichlet Allocation (LDA) (**?**) already used the *Simple Pólya urn model*, its generalized version proposed in Mimno et al. (**?**) allows to incorporate word relatedness directly into the inference process. Hence, a simple extension would be evaluating word relatedness based on word embeddings for the *Generalized Pólya urn model*. The topic shifting problem can be mitigated by jointly considering the global and local context of a word: if two terms appear in similar context-windows but do not share similar global contexts (i.e. corpus themes), they probably convey different topics.

In this paper, we propose a Context-aware Pólya urn model (Context-GPU) to generate topics by extracting topical phrases combining the local and global context of words/phrases[1]. We first detect the medical phrases in clinical reports by means of an off-the-shelf medical concept extraction tool; hence, the phrases extracted are thus reliable and clinically relevant. Then we use a modified Generalized Pólya urn model, which promotes words/phrases under the same topic if they are close neighbors in the window-based embedding (local context) as well as in the corpus-based embedding (global context) space. The window-based embedding improves the capability to detect semantic relatedness at the phrase level; also, it encodes word co-occurrences from an external source of knowledge (e.g. Wikipedia) alleviating the lack of statistics for technical terms. Simultane-

---

[1]https://github.com/gabrer/context_gpu/

ously, the corpus-based embedding limits the topic shifting during topic inference. To the best of our knowledge, this is the first time local and global context are combined for topical phrases extraction. Our experimental results have shown the effectiveness of this approach outperforming the previous methods in terms of quality of topics, topic coherence and efficiency.

We proceed to describe the related work. We then give a background of the Pólya urn model before presenting the proposed approach. Finally, we discuss our experimental results in comparison with the state-of-the-art approaches to topical phrase extraction.

Related work

Our work is related to three lines of research, phrase embedding learning, topic modeling incorporating word embeddings and using latent topics for language model learning.

Phrase Embedding Learning

Distributional semantic models (i.e. word embeddings) have recently been applied successfully in many NLP tasks (**?**). Neural network based approaches have become more efficient, allowing their use in multiple scenarios, thanks to the *skip-gram with negative-sampling training method* (SGNS), (**?**; **?**). It was widely popularized via *word2vec*, a software to create word embeddings. Recently, a new word embedding method has been proposed, called *FastText* (**?**), which treats each word as made of character n-grams. Vector representations are then computed from the sum of their n-gram representations. More traditional vector representations are based on a dimensionality reduction obtained by applying the Singular Value Decomposition (SVD) to the weighted document-term matrix of the corpus; Latent Semantic Analysis (LSA) (**?**) is a prominent method following this approach.

Phase embeddings can be simply taken as the average of their constituent word embeddings. If treating each phrase as a single term, its representation can also be learned from data directly using word representation learning methods such as LSA, SGNS or FastText. There have also been compositional semantic models which aim to build distributional representations of a phrase from its constituent word representations using Convolutional Neural Networks (CNNs) (**?**), based on features that capture phrase structure and context (**?**) or using convolutional tensor decomposition (**?**).

Topic Modeling Incorporating Word Embeddings

To exploit the information encoded into word embeddings, several models have been proposed combining topic models and word embedding representations. Gaussian LDA (**?**), for instance, use pre-trained word embeddings learned from large corpora (e.g., Wikipedia) to model topics as Gaussian distributions over the vector representations, defining topics as random samples from a multivariate Gaussian distribution whose mean is the topic embedding.

Nguyen et al. (**?**) proposed to use the word embeddings pre-trained from an external large corpus as latent word features to define categorical distributions over words, which is called a latent feature component. The original topic-to-word Dirichlet multinomial component in LDA which generates the words from topics is then replaced by a two-component mixture of the original Dirichlet multinomial component and a latent feature component. But model learning is difficult because of the coupling between the two components.

An alternative approach is TopicVec (**?**) which replaces the multinomial topic-word distribution with a probability function, it computes a focus word from a topic and word neighbors within the embedding; in TopicVec this link function is in addition combined with a context word embeddings along with the topic embedding and the focus word embedding.

Li et al. (**?**) measured the word relatedness based on pre-trained word embeddings and used it to modify the Gibbs sampling inference in a generalized Pólya urn model; overall, this strategy significantly reduces the computational cost compared to the aforementioned approaches. However, not only it is fully focused on the short-text analysis (i.e. one document one topic), but it doesn't exploit any global context to mitigate the topic shifting issues induced by word embeddings. Also, it didn't explore the benefit of using a n-gram word embedding such as FastText against the word-oriented embeddings.

Using Latent Topics for Language Model Learning

While the aforementioned approaches incorporate the word embeddings into topic model learning, there have also been attempts making use of latent topics to improve language models. Dieng et al. (**?**) proposed TopicRNN in which the global semantics come from latent topics as in typical topic modeling, but local semantics is defined by the language model constructed using Recurrent Neural Networks (RNNs). The separation of global vs local semantics is achieved using a binary decision model for stop words. Topic vectors here are also sampled from a Gaussian distribution with zero mean and unit variance and are refined during language model learning. In a similar vein, Lau et al. (**?**) proposed a topic-driven neural language model that also incorporates document context in the form of latent topics into a language model implemented using Long Short-Term Memory (LSTM) networks. They essentially treated the language and topic models as subtasks in a multi-task learning setting, and trained them jointly using categorical cross-entropy loss.

Pólya urn Models

In this section, we give a background of both simple and generalized Pólya urn Models. We describe how they can be used for topic extraction, before presenting in the next section, our proposed approach that extends them to exploit word contexts.

As shown in Mimno et al. (**?**), simple LDA model might not be able to fully capture the already available statistics of word co-occurrences in a corpus. Detecting semantic similarity between words is challenging due to the power-law characterization of natural language, i.e., words sharing a

common semantic might rarely co-occur together and hence being overlooked. A more effective model called Generalized Pólya urn model was proposed in Mimno et al. (**?**), by extending the Simple Pólya urn model used in LDA where the topic-word component is updated in order to strengthen the associations between related words under the same topic.

Simple Pólya Urn Model

The generative process of LDA can be interpreted by means of Pólya urn model (**?**), a statistical model describing objects of interest (e.g. words or topics) in terms of colored balls and urns.

In the context of topic models, balls can be considered as words and urns as topics; in particular, LDA follows the so-called *Simple Pólya urn* (SPU) model. In the main step of this process, a colored ball is randomly drawn from an urn and is put back along with an additional new ball of the same color; this induces a self-reinforcement process known as "rich get richer", since the probability of seeing a specific colored ball from an urn increases every time this ball has been drawn.

Likewise, LDA follows the SPU model by employing two kinds of urns: topic-document and word-topic urns. The topic-document urns hold balls whose color corresponds to different topics in a document, while the balls in the word-topic urns represent different words in a topic. The generative process proceeds as follows: a ball is extracted from the topic-document urn d_m, and its color determines the new topic assignment $\hat{z}$, then the ball is put back along with another ball of the same color. Next, a ball is extracted from the word-topic urn $\hat{z}$ determining a new word $\hat{w}$ and, as before, the ball with an additional one of the same color is put back into the urn. As a result, both the topic $\hat{z}$ and the word $\hat{w}$ increase their proportion in the topic-document and word-topic distribution, respectively.

Generalized Pólya Urn Model

The described process is intrinsically biased to promote together words that frequently occur in a corpus, overlooking less prominent but correlated words. To alleviate this shortcoming and increase the association strength between rare but still related words a *Generalized Pólya Urn* (GPU) model was proposed by Mimno et al. (**?**). It incorporates a corpus-specific word co-occurrence metric into the generative process affecting the probabilities of related words under the same topic.

Unlike the aforementioned simple version, in a generalized Pólya urn model when a ball of color $\hat{w}$ has been drawn, A_{vw} additional balls of several colors $v = \{1, ..., W\}$ are placed into the urn. This process increases, not only the probability of the observed word $\hat{w}$, but also the probability of its related words, and is commonly referred as *promotion* of the colored balls (**?**). Specifically, the LDA inference process now relies on a modified Gibbs sampling algorithm which simultaneously increases the probability of a word and their correlated terms at each iteration. Word relatedness is computed by weighting word co-occurrences using the standard Inverse-Document Frequency (IDF) weighting strategy $\lambda_v = log(D/D(v))$, where D is the number of documents and $D(v)$ is the number of document where the word v occurs at least once; this weight has the beneficial property of being higher for rare words increasing their prominence.

However, the effectiveness of this approach strongly depends on how accurately word correlations are identified. Although the GPU framework proposed by Mimno et al. (**?**) has improved the average quality of mined topics, it still relies exclusively on the global context of words (i.e. word co-occurrences in the corpus) and might completely overlook sentence-specific meaning of a word conveyed by the word's local context.

This drastically narrows the model capability to deal with multiple sense of words. For example, looking at the sentences "*White blood cell count is low.*" and "*This raises the blood sugar back to its normal level.*", current model might put under the same topic words like "*cell*" and "*sugar*", which are rather unlikely to appear coupled in a sentence. Moreover, similar issues can be experienced analyzing documents characterized by technical jargons which occur few times in corpus (i.e. poor statistics) and might exhibit a peculiar meaning for every phrase (i.e. multiple meanings).

Context-Aware Pólya Urn (Context-GPU) Model

In this section, we propose a modified Gibbs sampling algorithm to conduct a context-driven inference to cope with the described limitations. It exploits a word representation based on general knowledge source which provides a rich word statistics, and takes into account simultaneously the local and global context of words to disambiguate topically-irrelevant terms.

Our hypothesis is that the Generalized Pólya Urn model can be modified and enhanced to provide a framework combing local and global context of words. Local context is determined by a word embedding based on context window and trained on a large source of general knowledge (e.g. Wikipedia). Rather, the global context lies on the word representations obtained considering the term co-occurrences within a corpus. As a result, both local and global context can be incorporated into a context-aware Pólya urn model called *Context-GPU*, a generative model which is able to capture the semantics of a word with regard to both sentence and document context, mitigating the topic shifting issue induced by word embeddings.

Before presenting our proposed Context-GPU, we first describe how we extract medical phrases from clinical documents.

Medical Phrase Extraction

Medical terms in clinical documents are often expressed in multi-word phrases, for example, "*arterial blood gas*" and "*heart transplant*". These phrases are not semantically decomposable, as once split into unigrams, they would lose their original semantic meanings.

We use an open source clinical annotation tool *MedTagger*[2] which extracts and annotates concepts from clinical re-

---

[2]http://ohnlp.org/index.php/MedTagger

ports by leveraging knowledge bases, machine learning and syntactic parsing. The output of MedTagger provides detailed information about the medical concept detected such as attributes, uncertainty, semantic group (i.e. Diagnosis, Test and Treatment) and so on. Also, it has achieved the-state-of-the-art performance in terms of F-Measure (0.84) at the *i2b2 NLP challenge* on the concept mention task (**?**).

Clinical reports are also characterized by many occurrences of medical abbreviations to favor brevity due to a large amount of information that needs to be synthesized in a short time and limited space. Detection of medical concepts through MedTagger is not only much more reliable than other general purpose techniques for phrase extraction, but allows also to effectively detect and preserve the medical abbreviations. Once medical phrases are detected, they are represented by compound words where constituent words are joined together by an underscore. E.g. words that compose the phrase "*short of breath*" are substituted by the compound word "*short_of_breath*", "*saphenous vein graft*" by "*saphenous_vein_graft*", and so on.

However, treating phrases as compound words leads to more severe data sparsity since phrases sharing common lower-order n-grams, such as "*right coronary artery*" and "*left coronary artery*" would be considered as two totally different terms. Also, preserving both the multiple words and the compound phrase is not a solution, as the phrases are naturally less frequent than individual words and would be ranked with lower probabilities.

To this end, once multiple words are substituted by a compound word, in the Context-GPU we adopt the FastText embedding (**?**), a word embedding oriented by design to deal with sub-grams composing words. Thus, it naturally fits the need to detect the similarity between a phrase and its constituent words. For example, in our trained FastText embeddings the word *saphenous_vein_graft* has neighbors such as *saph*, *aphe*, but also *vein* and *graft*. Therefore, we combine FastText with the Pólya urn model to increase the probability to see under the same topic the words *vein* or *graft* once we come across the phrase *saphenous vein graft*, and vice versa.

Local Context

Some commonly used embedding have been the SVD (**?**) and SGNS (i.e. word2vec) (**?**), and only recently FastText (**?**), and they all provide vectors encoding both syntactic and semantic information about a word at its local context window in a large corpus.

Two characteristic features of these embeddings are here exploited. The first is that words are represented by a vector trained with regard to the local contexts where the words are likely to appear. Therefore, it can be used to reinforce ties among words sharing common uses in phrases (e.g. *alzheimer* and *schizophrenia*). The second feature is that word embeddings are commonly trained on a large external source of data (e.g. Wikipedia), hence they can mitigate the low statistics of infrequent technical jargons or rare words in a corpus.

Words are considered related based on the geometric proximity of their vector representations. We propose two strategies to extract related words: a threshold and a Top-N approach. For the threshold approach, the related words of a given word are those whose cosine distances with the target word are less than a pre-defined threshold. Alternatively, the Top-N approach extracts a fixed number N of the closest words regardless of their actual distances. In the former approach, the number of neighbors is not fixed for different words, while in the latter the number is fixed but also unrelated words could be added to the neighbor set.

Global Context

Although topics extracted by combining word embedding and Pólya urn model are more consistent with the occurrence pattern of words in sentences, word embeddings have some well-known shortcomings related to antonyms (e.g. *tall* and *short*) or co-hyponyms (e.g. *schizophrenia* and *alzheimer*). Indeed, these are words that might share a similar context windows and then be potentially treated as directly correlated into the embedding space. To avoid any topic shifting resulting from word ambiguities, we balance the local context information with the corpus-specific context computed by applying the Latent Semantic Analysis (LSA) (**?**).

In particular, we use LSA to learn latent topics from data by performing Singular Value Decomposition (SVD) on the $V \times D$ term-document count matrix where V is the vocabulary size and D is the number of documents. SVD factorizes such a matrix into the product of three matrices, W, Σ, and C^{T}. In $W \in \mathbb{R}^{V \times m}$, each row represents a word and each column represents a dimension in a latent space which is orthogonal to each other. Σ is a diagonal $m \times m$ matrix which contains singular values along the diagonal indicating how important each latent dimension is. In $C^{\mathsf{T}} \in \mathbb{R}^{m \times D}$, each row represents one of the latent dimensions and each column represents a document. If taking the top k latent dimensions in W, we will have a reduced matrix $W_k \in \mathbb{R}^{V \times k}$ where each word is essentially represented by a dense k-dimensional vector. Hence, using LSA, we will be able to generate another set of word embeddings based on global context. For each word, we can then retrieve its related words using the thresholding or Top-N approaches mentioned above.

One may argue that topic models such as LDA already captures the global context information by compressing the original document into a lower-dimensional bag-of-topics representation. It is worth noting that LSA learns latent topics by performing SVD on term-document count matrix, and as a result, the topics are assumed to be orthogonal. LDA uses generative probabilistic models to generate latent topics which are represented as word distributions, and it uses Dirichlet priors for both the document-topic and topic-word distributions. In LDA, topics are allowed to be non-orthogonal. So although both LSA and LDA try to capture the global context, the topic results would be somewhat different. It has been pointed out previously that in some scenarios LSA outperforms LDA providing better quality topics (**?**). As will be shown in our experiments, additionally incorporating the global context derived by LSA into the context-aware Polya Urn model gives better performance.

Words are likely to express a common topic, not only

when sharing a common local context window (e.g. Fast-Text similarity), but also a global context (i.e. LSA similarity) depending on the analyzed documents. Therefore, we first extract the word neighbors both from local context and global context based embeddings, and we then preserve only the terms in the intersection of those sets, hence improving the probability that a word in a topic sharing both local and global context.

Topic Inference

Given the set of documents $\mathcal{D}$ and the topic assignments $\mathcal{Z}$, the conditional posterior probability of a word w in a topic z follows the standard generalized Pólya urn model (**?**):

$$P(w|z, \mathcal{W}, \mathcal{Z}, \beta, \mathbf{A}) = \frac{\sum_v N_{v|z} A_{vw} + \beta}{N_z + |\mathcal{V}|\beta} \qquad (1)$$

where $\mathbf{A}$ is a *promotion matrix* that expresses whether two words are related to each other, i.e., if one should influence the expectation to draw the other one.

The promotion matrix is critical for the overall algorithm performance, as it concisely expresses the available information about word relatedness. We propose to set the values of $\mathbf{A}$ by computing the word relatedness as a result of the $\mathcal{P}$ neighbors provided by the local context embeddings and the $\mathcal{Q}$ neighbors from the global context embeddings. For a word v, another word w is promoted if it is v's neighbor both at the local level (i.e., based on its local context embedding) and the global level (i.e., based on its global context embedding), as expressed in Eq. **??**. Thus, only if both words are correlated in both the local and global context embedding space, their corresponding cell value in $\mathbf{A}$ is updated to increase their probabilities to be drawn under the same topic. In the particular case of $\mathbf{A}$ being the identity matrix, the model collapse into the the Simple Pólya urn model, providing the posterior probability of a word w under a topic z in standard LDA.

The training procedure of our proposed context-aware Pólya urn model is shown in Algorithm **??**. The Gibbs sampling inference can be more complex and expensive due to the non-exchangeability property of words in the generalized Pólya urn model (i.e. under the same topic, joint probability of words is not invariant to permutation). Therefore, we follow the same approach adopted in Mimno et al. (**?**), considering each word as it was the last one during the inference process, ignoring the effect for subsequent words and their topic assignments.

$$A_{v,w} = \begin{cases} 1 & \text{if } w \in (\mathcal{P}_v \cap \mathcal{Q}_v) \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

Experiments

We assess the effectiveness of our proposed Context-GPU using the data released as part of the i2b2 Natural Language Processing Challenges for Clinical Records (**?**). The corpus consists of 1,243 de-identified discharge summaries, characterized by medical jargon, which describes medications, dosages, modes (e.g. oral, intravenous, etc.), frequencies, reasons for the treatment, and so on. Hence, we adopted this

---

Algorithm 1 Training procedure of the context-aware Pólya urn model.

Input: Corpus C, K topics, α, β, thresholds τ and σ
Output: Posterior topic-word distribution

```
 1: /* Medical phrase extraction */
 2: C_p ← MedTagger.PhraseDetection(C);
 3:
 4: /* Local and global neighbors */
 5: for v ∈ 𝒲 do
 6:     𝒫_v ← WindowEmbedding.Neighbors(v);
 7:     𝒬_v ← CorpusEmbedding.Neighbors(v);
 8: end for
 9:
10: /* Promotion matrix */
11: A_{v,w} ← ComputePromotionMatrix(𝒫_v, 𝒬_v)
12:
13: /* Generalized Pólya Urn sampling */
14: for d ∈ 𝒟 do
15:     for w_n ∈ w^d do
16:         N_{z_i|d_i} ← N_{z_i|d_i} − 1
17:         for all v do
18:             N_{v|z_i} ← N_{v|z_i} − A_{vw_i}
19:         end for
20:     end for
21:     sample z_i ∝ (N_{z|d_i} + α_z) (N_{w_i|z} + β) / (Σ_{z'} N_{w_i|z'} + β)
22:     for w_n ∈ w^d do
23:         N_{z_i|d_i} ← N_{z_i|d_i} + 1
24:         for all v do
25:             N_{v|z_i} ← N_{v|z_i} + A_{vw_i}
26:         end for
27:     end for
28: end for
```

---

dataset to assess the model efficacy to deal with multi-phrase concepts and domain-specific jargon.

Clinical reports are preprocessed by removing the English common stop words as well as the clinical stop words (e.g. "Dr.", "medical problem", "discharge", etc.). We filter out the most frequent 10 words and the words occurring less than 5 times. We use the MedTagger software to detect the medical phrases and substitute them within the documents. We do not perform stemming. As a result, in the "bag-of-words" representation the vocabulary size is 7,883, while in the "bag-of-phrases" increases to 9,932.

Word embeddings are trained on a Wikipedia 2015 snapshot combined with the i2b2 dataset. We use the *hyperwords* library[3] (**?**) to train the 300-dimensional SVD and SGNS embeddings, configured with the default parameters. Likewise, we train the 300-dimensional FastText embedding using the library provided by Facebook Research[4] and setting the n-gram sizes between 2 and 6. The LSA representation adopted as local context is computed on the i2b2 dataset; we use the S-space library[5] to obtain the final 300-dimensional

---

[3]https://bitbucket.org/omerlevy/hyperwords
[4]https://github.com/facebookresearch/fastText
[5]https://github.com/fozziethebeat/S-Space

representation of words and documents.

We train Context-GPU and set θ and σ to 0.7 and 0.8 respectively based on a grid search of values in $[0.5, 0.6, 0.7, 0.8, 0.9]$ using 5-fold cross validation. We set the maximum number of Gibbs sampling iterations to 1500. We compare Context-GPU with the following baselines:

- LDA. We use the LDA implementation in MALLET[6] with the default settings and perform hyperparameter optimization every 200 iterations.
- Generalized Pólya urn (GPU) model (**?**). We implemented this algorithm by modifying the LDA implementation in the MALLET library.
- TopicVec (**?**). We use the available implementation[7] with the default configuration, increasing the maximum iteration number.
- TPM (**?**). We implemented the Topical Phrase Model which extracts medical topics using both MedTagger and a hierarchy of Pitman-Yor processes. It outperformed other topical phrase extraction models.

Topic coherence

We assess the generated topics by evaluation of their topic coherence. We adopt the topic coherence measure proposed in Mimno et al. (**?**), which relies on the co-occurrence statistics collected from the analyzed corpus; this allows us to directly measure the coherence of topics with topical phrases (e.g. *short_of_breath*).
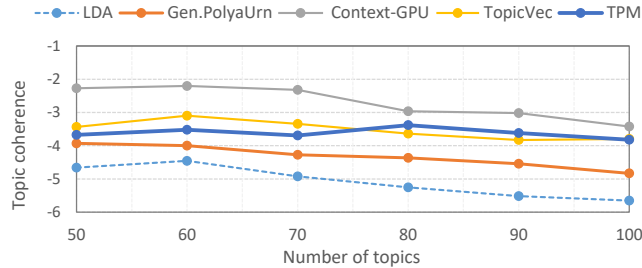

Figure 1: Topic coherences vs. number of topics.

In our evaluations, we compute the topic coherence on the top 10 words/phrases using the implementation provided in the Palmetto library[8] (**?**). In Figure **??** we report the topic coherence computed by averaging the coherences resulting for each topic. A peak of coherence is obtained around 60/70 topics for every model, suggesting a potentially suitable number of topics to discriminate the documents. GPU with only local context incorporated outperforms LDA, but its performance is worse compared to TopicVec or TPM. Context-GPU gives superior results over all the baseline models, in particular around 60 and 70 topics. This shows that additionally incorporating global context is essential to achieve better topic coherence results compared to only

---

[6] http://mallet.cs.umass.edu
[7] https://github.com/askerlee/topicvec
[8] https://github.com/dice-group/Palmetto

considering local context. Also, our proposed Context-GPU only involves simple modifications to GPU, but it appears to be more effective than more complicated ways of incorporating word embeddings into topics models (such as TopicVec) or assuming word generation following the HPYP process (such as TPM).

To extract topical phrases from text, we have explored a few different ways in learning word/phrase representations such as learning directly from our data using SVD, training a combined Wikipedia/clinical report data using SGNS or FastText. In Figure **??** we compare these word/phrase embedding learning results over our Context-GPU. We can observe that SVD and SGNS perform similarly in most cases and SVD even slightly outperforms SGNS when the topic number is set to 80 or 90. FastText outperforms the other two word/phrase embedding learning methods especially when the topic number is lower than 80. This shows that Fast-Text built on character n-grams is more effective in capturing phase sub-structures.
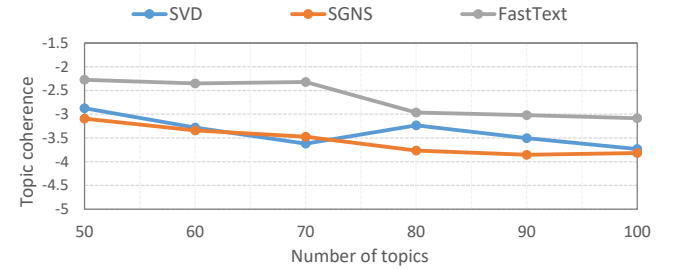

Figure 2: Context-GPU with different word/phrase embedding learning methods vs. number of topics.

Finally, we compare in Figure **??** the execution time needed to train the models, excluding the constant time required by each model to load the embeddings. We did not plot the training time for TPM in the figure as it required significantly more time (over 12 hours) compared to all the other models, showing that modeling phrase generation using HPYP is very expensive. For the remaining models, TopicVec is computationally more complex than the others. Both GPU and Context-GPU have no noticeable difference and they both required three-fold the training time of LDA. Overall, Context-GPU appears to be more effective compared to TopicVec and TPM.
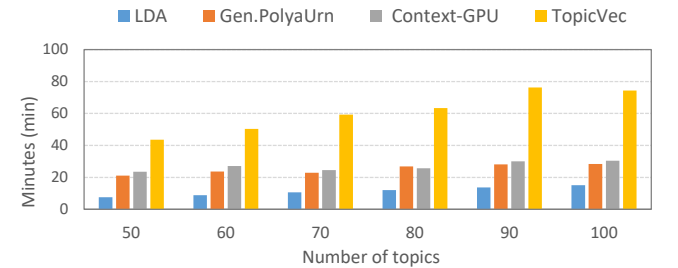

Figure 3: Execution time vs. number of topics.

Topic 1	Topic 2	Topic 3	Topic 4
		TopicVec	
carotid	diuresis	dyspnea on exertion	congestive heart failure
coronary artery	torsemide	ejection fraction	fibrillation
magnesium	cardiomyopathy	pulmonary	ejection fraction
saphenous vein graft	shortness of breath	atrial fibrillation	insufficiency
potassium chloride	torsemide 100 mg	diuresed	calcium
coronary artery bypass grafting	spironolactone 25 mg	congestive heart failure	intubation
mitral insufficiency	diuretic	ischemia	thyroid
mitral regurgitation	aldactone	diabetes mellitus	vascular congestion
potassium	pleural effusion	propafenone	tricuspid regurgitation
substernal	pulmonary edema	volume overloaded	right knee
		Contex-GPU	
pregnancy	mitral regurgitation	coronary artery disease	congestive heart failure
ultrasound	digoxin	cardiac transplant	pulmonary edema
postpartum hemorrhage	pleural effusion	cardiomyopathy	orthopnea
endometrial biopsy	orthopnea	right coronary artery	nonischemic
total abdominal hysterectomy	dilated cardiomyopathy	pravachol 20 mg	diastolic dysfunction
postpartum	plavix 75 mg	paroxysmal atrial fibrillation	cardiomyopathy
vomiting	shortness of breath	cyclosporine	heart failure
salpingo oophorectomy	dyspnea on exertion	herpes zoster	shortness of breath
physical examination	tachyarrhythmia	fenofibrate tricor	cardiac catheterization
fibroid	pulmonary edema	right coronary artery	atrial fibrillation

Table 1: Topics generated by TopicVec and Context-GPU in 70-topic runs.

Topic Qualitative Assessment

We report in Table **??** some topics generated in a 70-topics run. For the sake of brevity, we report only the topics of TopicVec and Context-GPU since TopicVec gives similar coherence scores as TPM but requires significantly less training time compared to TPM. TopicVec inference learns both word and topic embeddings simultaneously. It allows the model to take into account the local context of words, which in turn, alleviates the lack of global statistic for a term. Both the topics of TopicVec and Context-GPU are able to generate topical phrases. However, in several topics of Context-GPU, we can distinguish a gradual definition of the analyzed themes, which reflect better semantic coherence. For example, in Topic 4, it can be observed a gradual topic refinement under Context-GPU from the general purpose terms such as *felt* or *insufficiency* to more characterizing words/phrases such as *shortness of breath*, *atrial fibrillation*. In addition, we can observe under the same topic symptom and medication, such as *cardiomyopathy* and *plalix 75 mg*. As a result, the overall expressiveness of topics extracted by the Context-GPU outperforms TopicVec, both thanks to their internal coherence and to the improved expressiveness of the adopted words/phrases.

Conclusion

We have described a new approach which aims to effectively combine the local and global context of words and phrases. It first detects high reliable phrases and then generates topics using our proposed Context-aware Pólya urn model. This statistical model combines the word semantic encoded by the context-based and corpus-based embeddings. In particular, we have exploited the LSA and FastText embeddings. The former improved the ties of a word to the corpus themes; the latter allowed a fine-grained use of a word depending on the phrase in which it occurs. An experimental comparison with the state-of-the-art methods has shown an improved coherence of final topics and a decreased computational cost.

Magni tempora pariatur necessitatibus odit incidunt similique reprehenderit voluptatibus reiciendis maxime, asperiores possimus temporibus sapiente iure voluptate minus quae quaerat provident recusandae, officiis dicta doloribus, numquam amet officiis repudiandae quod corrupti?Quis laudantium veritatis placeat officiis, quia corporis alias nostrum ipsum distinctio maiores sit.Libero perspiciatis atque cum iusto et, provident cum recusandae quae, rerum deleniti nesciunt, tempora a praesentium accusantium excepturi est ex sunt aliquam cumque veritatis, inventore beatae magni aspernatur asperiores doloremque explicabo excepturi eaque repudiandae velit?Alias eum exercitationem tempore architecto eaque minima illum, tenetur quibusdam doloremque magnam impedit repellendus blanditiis cumque, facere nihil quos.Natus non quo numquam corporis, et consectetur incidunt natus rerum tempora laborum, iure debitis laboriosam qui ut tempore at quisquam minima, saepe veniam corporis nesciunt facilis sunt reprehenderit, cupiditate consectetur as-

sumenda aliquid est unde.Error esse eos, vitae repellendus iusto beatae veritatis ad sapiente perferendis nesciunt dolorum quam voluptate, ratione magni provident sapiente cupiditate optio dicta,