CS388 NLP Project Report

Text Mining for Hidden Relations and Trending

C. Vic Hu
vic@cvhu.org

Ali Unwala
aliunwala@gmail.com

***Abstract*—The main contribution of this paper has two folds. First, we formalized an illustration method to visualize topic trending on a bidirectional spanning tree, which gives a meaningful intuition to discover how hidden thematic structures in large archive of text documents change, merge and split over time. Second, we proposed an algorithm, Thematic Particle Clustering, that combines probabilistic sampling, clustering and gradient descent methods to predict upcoming topics based on a sequence of history topics. The effectiveness of our methods is demonstrated through a collection of 10,000 patent data in the field of robotics spanning over 30 years.**

I. Introduction

In this paper we address two problems. The first problem that we encountered was that it is hard to understand topics as they changed over time. Topics intrinsically create an twisted structure over the course of time. To solve this we made an visualization technique for topics that shows topics converge from year to year and new topics forming. To do this topics from year to year are superimposed onto each other to create a tree structure. This allows us to see what topics converged and how fast they converged over time. We found this style of visualization novel and immensely useful in understanding topic relationships from year to year as they progresses.

The second problem we solved was current topic modeling over time does not add Gaussian noise to topics. Allowing noise allows a topic to take a new path over time. We are able to do this by representing topics as vectors in an N dimensional word space. Once each topic looks like a vector we can add noise an perform ***CLUSTERING?**

We evaluate our approach to this second problem by allowing

II. Background

Given a collection of text-based patent documents, one intuitive idea to find out trending patterns is to examine the underlying thematic structures hidden in the text. Based on the vocabulary distribution, we want to know what the intrinsic topics are implied in the given context. One common way to do exactly that is the Probabilistic Topic Models formalized by Blei et al. [1]

A. Probabilistic Topic Models

The main objective of topic modeling is to automatically discover the unobserved hidden structures–the topics, per-document topic distributions, and the per-document per-word topic assignments, while a collection of text documents is the only observable variables.

A mounting bracket mounts a photovoltaic module to a support structure.
Claims What is claimed: 1. A mounting bracket comprising: a bottom flange; an upright portion extending from the bottom flange and having an inner surface and an outer surface; a top flange opposite the bottom flange, extending from the upright portion and having a downward facing inner surface configured to adjoin an upper surface of a photovoltaic module; a first extension extending from the inner surface of the upright portion at a position between the top flange and the bottom flange and having a first surface that defines a first groove sized to accommodate an edge of the photovoltaic module with the downward facing inner surface of the top flange and a second surface opposed to the first surface; a second extension adjacent to the first extension and extending from the

Fig. 1: A sample patent document (partial)

For example in Fig. 1, we have annotated a selection of words, with topics distinguished by colors. For the orange topic, we get words like flange, surface and extending, which could be interpreted as the attachment of hardware components. Similarly, the blue and green topics could be translated into topics about installation and mounting respectively. By looking at the text, most human being with common comprehensibility could easily tell what a patent data like Fig. 1 is about, and accordingly highlight the relevant keywords that compose such topics.

Nonetheless, the efficiency and accuracy of human labors don't scale up easily when the size or complexity of these patent documents increases. The objective of probabilistic topic modeling is to automate this inference process and to provide hidden insights and meaningful intelligence of big data. If we are able to successfully construct a probable thematic structure from a large archive of text data for each time slice in a sequence, we could presumably infer how these topics inherit or inspire each other, and most excitingly, predict the most likely topics in the future.

B. Latent Dirichlet Allocation

Latent Dirichlet allocation, or LDA, is the simplest topic model [2] that assigns each word in the documents a distribution over a fixed number of topics. Instead of having a hard boundary between topic collections, LDA provides a distribution of topics per document, giving the likelihood of a mixed proportion of topic assignments. Namely, all text documents share the same set of topic collection but with different proportions to each topic. For instance in Fig. 2, although there are $K = 100$ topics overall, only a few topics were actually activated.

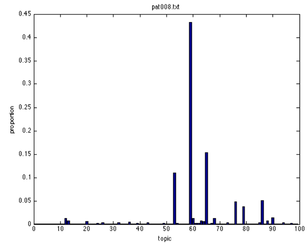

Fig. 2: A sample topic proportion of a patent

$\beta_k,\ k = 1 \cdots K$	The K topics, represented by a distribution over words.
$\theta_d,\ d = 1 \cdots D$	Topic proportions for document d, where $\theta_{d,k}$ is the topic proportion of topic k for document d.
$z_d,\ d = 1 \cdots D$	Topic assignments for document d, where $z_{d,n}$ is the topic assignment for the n-th word in document d.
$w_d,\ d = 1 \cdots D$	The observed words for document d, where $w_{d,n}$ is the n-th word in document d.

TABLE I: Topic modeling notations

To build the generative probabilistic model, we compute the joint distribution and use it to estimate the posterior probability. With the notation specified in Table. I, the LDA generative process can be formalized as the following joint probability of both hidden and observed random variables:

$$
\begin{aligned}
&p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) \\
&= \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\beta_{1:K}, z_{d,n}) \right)
\end{aligned}
$$

which can also be expressed as a graphical model:

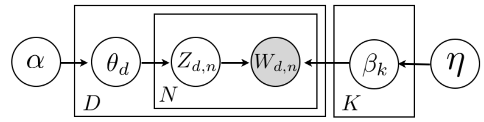

Fig. 3: LDA graphical model. Nodes represent variables, while edges indicate the dependency relations. The shaded node is the only observed variable (document words), and all others are the hidden variables. The D plate denotes the replicated variables product over D documents, while the N plate denotes replication over N words in each document.

Note that there are several conditional dependencies implied in the graphical models, which reflects the main principles of how LDA "think" the documents are generated:

1) Randomly pick a distribution θ_d over topics.
2) For each word in the document
   a) Randomly choose a topic from the previously-chosen distribution $\theta_{d,n}$.
   b) Randomly choose a word from the corresponding distribution $Z_{d,n}$.

Assuming this generative process is how our documents are created, now LDA uses the graphical model in Fig. 3 to infer the posterior probability of the hidden structures given our observable:

$$
p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})}
$$

The computation of possible topic structures is often intractable and the posterior distribution can only be approximated in most cases. To form an approximation algorithm, topic modeling can generally be categorized as sampling-based algorithms and variational algorithms. The most popular sampling method for topic modeling is Gibbs sampling, which introduces a sequence of random variables to construct a Markov chain and collects samples from the limiting distribution to estimate the posterior. Instead of using samples to approximate the posterior, variational methods find the closest parameterized distribution candidate by solving optimization problems [2] [3].

C. Limitations & Potential Improvements

Although LDA provides a powerful perspective to browsing and interpreting the implicit topic structures in our patent corpus, there are a few limitations it imposes against further discoveries. An extensive amount of research has been focused on relaxing some of the assumptions made by LDA to make it more flexible and suitable for various adaptations in more sophisticated context.

LDA is essentially a bag-of-words probabilistic model. Namely, it constructs a word-frequency vector for each document but disregards the word ordering and the neighboring

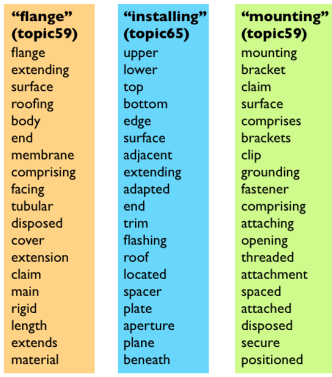

Fig. 4: The top 3 topics of a sample patent

context. Although this assumption looses the syntactic information and sometimes seems unrealistic when processing natural language, it is usually good enough when capturing the document semantics and simplifying hidden structural inferences. Nonetheless, for more sophisticated tasks such as language generation or writing style modeling, the bag-of-words assumption is apparently insufficient and needs to be relaxed. In these cases, there are variants of topic models that generate topic words conditioned on the previous word [4], or switches between LDA and hidden Markov models (HMM) [5].

The LDA graphical model in Fig. 3 is invariant to the ordering of our patent documents, which could be inappropriate if the hidden thematic structure is actually dependent on sequential information such as years published, which is typical in document collections spanning years, decades or centuries. To discover how the topics change over time, the dynamic topic model [6] treats topics as a sequence of distributions over words and tracks how they change over time.

In either LDA or more sophisticated dynamic topic models [6], the number of topics $\beta_{1:K}$ is determined manually and assumed to be fixed. One elegant approach provided by the Bayesian nonparametric topic model [7] is to find a hierarchical tree of topics, in which new documents can now imply previously undiscovered topics.

To include additional attribute information associated with the documents such as authorships, titles, geolocation, citations and many others, an active branch of research has been performed to incorporate meta-data in topic models. The author-topic model [8] associates author similarity based on their topic proportions, the relational topic model [9] assumes document links are dependent on their topic proportion distances, and more general purpose methods such as Dirichlet-multinomial regression models [10] and supervised topic models [11].

Many other extensions of LDA are available, including the correlated topic model [12], pachinko allocation machine, [13], spherical topic model [14], sparse topic models [15] and bursty topic models [16].

III. Problem Definition and Algorithm

A. *Task Definition*

B. *Maximum Path Branching Model*

C. *Thematic Particle Clustering*

Built on top the results obtained from the LDA topic models, our Thematic Particle Clustering (TPC) algorithm aims to make topic predictions for the upcoming year based on what it has seen in the past. Our goal is to formalize a set of particles $\mathbf{w}_{1:N}$ inferred from the topics from previous years, cluster them and use the results to describe what we think the upcoming topics $\beta_{1:K}$ will be. Before jumping into the details of TPC, let's go over some of the fundamental concepts and definitions we will use.

1) Distance Functions: The idea of a particle is essentially a sampled instance of the topic distribution over a time sequence, represented by a vector $\mathbf{w}_i \in \mathbb{R}^{|V|},\ i \in \{1, \cdots, N\}$, where $|V|$ is the total vocabulary size of all the topic words appeared. Since one of our intermediate objectives is to formalize clusters between these particles, we need to first define how we will measure the similarity or distance between any pair of particles $\mathbf{w}_i,\ \mathbf{w}_j$.

Minkowski:

$$d = \sqrt[p]{\sum_{k=1}^{|V|} |w_{ik} - w_{jk}|^p}$$

Note that when $p = 1$, the Minkowski reduces to the city block distance, while $p = 2$ gives the Euclidean distance and $p = \infty$ yields the Chebychev distance.

Cosine:

$$d = 1 - \frac{\mathbf{w}_i\,\mathbf{w}_j^T}{|\mathbf{w}_i|_2|\mathbf{w}_j|_2}$$

Correlation:

$$d = 1 - \frac{(\mathbf{w}_i - \overline{\mathbf{w}}_i)(\mathbf{w}_j - \overline{\mathbf{w}}_j)^T}{|(\mathbf{w}_i - \overline{\mathbf{w}}_i)|_2|(\mathbf{w}_j - \overline{\mathbf{w}}_j)|_2}$$

where

$$\overline{\mathbf{w}}_i = \frac{1}{|V|}\sum_{k=1}^{|V|} w_{ik},\ \overline{\mathbf{w}}_j = \frac{1}{|V|}\sum_{k=1}^{|V|} w_{jk}$$

Jaccard:

$$d = \frac{\#\left[(w_{ik} \neq w_{jk}) \cap ((w_{ik} \neq 0) \cup (w_{jk} \neq 0))\right]}{\#\left[(w_{ik} \neq 0) \cup (w_{jk} \neq 0)\right]}$$

2) Clustering Algorithms:

Assign K topics to N particles uniformly;
Add Gaussian noise to particles;
Cluster particles into K groups (TF-IDF weights with cosine/Euclidean distances);
Compare the clusters with topics from the next year, apply discounts to current weights, and adjust to new weights;
repeat;

3) Putting It Together:

IV. Experimental Evaluation

A. Methodology

add the baseline method (comparing bigram errors)

B. Results

C. Discussion

V. Related Work

VI. Future Work

VII. Conclusion

References

[1] Blei, D. Introduction to Probabilistic Topic Models. Princeton University. 2011.

[2] Blei, D., Ng, A. and Jordan, M. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993-1022, January 2003.

[3] Hoffman, M., Blei, D. and Bach, F. On-line learning for latent Dirichlet allocation. In Neural Information Processing Systems, 2010.

[4] Wallach, H. Topic modeling: Beyond bag of words. In Proceedings of the 23rd International Conference on Machine Learning, 2006.

[5] Griffiths, T., Steyvers, M., Blei, D. and Tenenbaum, J. Integrating topics and syntax. In L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, pages 537-544, Cambridge, MA, 2005. MIT Press.

[6] Blei, D. and Lafferty, J. Dynamic topic models. In International Conference on Machine Learning, pages 113-120, New York, NY, USA, 2006. ACM

[7] Teh, Y., Jordan, M., Beal, M. and Blei, D. Hierarchical Dirichlet process. Journal of the American Statistical Association, 101(476):1566-1581, 2006.

[8] Rosen-Zvi, M., Griffiths, T., Steyvers, M. and Smith, P. The author-topic model for authors and documents. In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, pages 487-494. AUAI Press, 2004.

[9] Chang, J. and Blei, D. Hierarchical relational models for document networks. Annals of APplied Statistics, 4(1), 2010.

[10] Mimno, D. and McCallum, A. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In Uncertainty in Artificial Intelligence, 2008.

[11] Blei, D. and McAuliffe, J. Supervised topic models. In Neural Information Processing Systems, 2007.

[12] Blei, D. and Lafferty, J. A correlated topic model of Science. Annals of Applied Statistics, 1(1):17-35, 2007.

[13] Li, W. and McCallum, A. Pachinko allocation: DAG-structured mixture models of topic correlations. In International Conference on Machine Learning, pages 577-584, 2006.

[14] Reisinger, J., Waters, A. ,Silverthorn, B. and Mooney, R. Spherical topic models. In International Conference on Machine Learning, 2010.

[15] Wang, C. and Blei, D. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, Advances in Neural Information Processing Systems 22, pages 1982-1989. 2009.

[16] Doyle, G. and Elkan, C. Accounting for burstiness in topic models. In International Conference on Machine Learning, pages 281-288. ACM, 2009.