Probabilistic Text Analytics Framework for Information Technology Service Desk Tickets

Ea-Ee Jan
IBM T.J. Watson Research Center
ejan@us.ibm.com

Kuan-Yu Chen
IBM T.J. Watson Research Center
kychen@iis.sinica.edu.tw

Tsuyoshi Idé
IBM T.J. Watson Research Center
tide@us.ibm.com

***Abstract*—Ticket annotation and search has become an essential research subject for the successful delivery of IT operational analytics. Millions of tickets are created yearly to address business users' IT related problems. In IT service desk management, it is critical to first capture the pain points for a group of tickets to determine root cause; secondly, to obtain the respective distributions in order to layout the priority of addressing these pain points. An advanced ticket analytics system utilizes a combination of topic modeling, clustering and Information Retrieval (IR) technologies to address the above issues and the corresponding architecture which integrates of these features will allow for a wider distribution of this technology and progress to a significant financial benefit for the system owner. Topic modeling has been used to extract topics from given documents; in general, each topic is represented by a unigram language model. However, it is not clear how to interpret the results in an easily readable/understandable way until now. Due to the inefficiency to render top concepts using existing techniques, in this paper, we propose a probabilistic framework, which consists of language modeling (especially the topic models), Part-Of-Speech (POS) tags, query expansion, retrieval modeling and so on for the practical challenge. The rigorously empirical experiments demonstrate the consistent and utility performance of the proposed method on real datasets.**

Keywords—Analytics, ITIL, IT operations, Ticket, IT service, topic modeling, information retrieval

I. INTRODUCTION

The IT service desk consists of a supporting set of infrastructure, processes and services that assist company's employees and business users for any IT related problems. The issues that the service desk typically handle include password reset, access issues, application issues, firewall not working, how to setup mail box, and so on. Some of the issues are tied to personal configurations, related to "how to" and "request for information", or asked to servers and systems. An issue reported to the service desk is treated as an IT service desk ticket. Unlike data center IT delivery tickets [7, 13], which mainly address hardware issues with less human interaction, such as servers operations, managements, and applications configuration issues, IT service desk tickets focus more on users' issues and process implementations.

In enterprise systems, millions of IT service desk tickets are needed to be processed per year. For effective IT service desk operations, it is critical to know what the major ticket drivers are, whether there are repeating patterns, how many tickets can be used for an automated solution, and how can we improve the productivity of the service desk agents. These issues all come down to three key questions: What are the pain points? How many tickers are related to each pain points? Is there an automatic recommendation to address an open ticket?

Today, IT tickets are managed by the Incident-Problem-Change (IPC) ticket system and may follow the ITIL (*Information Technology Infrastructure Library)* process standard. Structured and unstructured data are stored in a transactional database for ticket management and analysis. However, it is obvious that the quality of data in the transactional system is very low. The possible reasons are two-fold. First, due to various factors in practice, such as time pressure and growing back log, structured fields contain many missing entries as well as abbreviated or inaccurate pieces of information. Second, in the descriptive text or unstructured data fields, data elements are written by human agents to address their customers concerns in a hurry. As a result, typos, spontaneous abbreviation, grammatical errors, templates attached with an agent's conversation, addresses, or over length text cutoff are very common. Moreover, the IT service desk tickets contain many domain specific technical terms, properties and product names. These terms and names are seldom addressed in today's web anchors and hyperlinks. The web resources are less effective when applying to these tickets.

To address these challenges, the use of advanced text analytics is part of any IT Operational Analytics delivery. The major advantage of text analytics is that, with the aid of textual corpuses, it effectively removes uncertainties of the data to produce normalized information. However, due to the challenges we described above, naïve applications of existing text mining solutions do not work very well for the IT service desk tickets. Motived by the practical request, this paper presents a new and novel framework for analyzing IT service desk tickets on the ground of text analytics technologies.

II. TOPIC MODELING

The vector space model (VSM) [8] is the basis for most of the IR-related researches until now [3, 9]. However, the flaws of VSM are two-fold. On one hand, VSM might suffer from the word usage diversity, which sometimes degrades the performance severely as a given query and its relevant documents may use different sets of words (e.g., synonyms). On the other hand, lots of polysemy words have different meanings in different contexts. To complement the above drawbacks of VSM, LSA [8] assumes that there is an implicit semantic structure between words and documents, which can be explored by performing SVD on a pre-defined word-by-document matrix.

Instead of LSA, probabilistic topic models have been proposed as a counterpart of the non-probabilistic methods. The probabilistic Latent Semantic Analysis (pLSA) [16] and the Latent Dirichlet Allocation (LDA) [1] are two well-practiced representatives. Take pLSA for example, the probability of a word occurring in a document d will be reformulated by:

$$P(w_i|d) = \sum_{k=1}^{K} P(w_i|T_k, d)P(T_k|d) \approx \sum_{k=1}^{K} P(w_i|T_k)P(T_k|d), \quad (1)$$

where each document d is consisted of a set of K shared latent topics $\{T_1,\ldots,T_k,\ldots,T_K\}$ associated with the document-specific weights $P(T_k|d)$ and each topic T_k in turn offers a unigram distribution for observing an arbitrary word of the language.

III. The Proposed Framework

Most of topic models work with a bag-of-words assumption. In this paper, we extend the conventional approach to phrases, meaningful n-gram from vocabularies, to represent topics.

A. Text Normalization & Pre-Processing

To handle noisy text from IT service desk tickets, we developed text normalization pre-processors, including xml tags and stop words removal, stemming, and punctuations/abbreviation normalization. We also limit word length to remove email, http link and other functionless words. The "tab" and "carriage return" markers can be optional preserved to provide additional syntactic (structure) information.

B. Concept Analysis

Topic modeling can be used to digest word usage cues in each document. We assume that each latent topic conveys some ideas which are common to a subset of the input data. To better visualize/understand each topic, we want to represent topics by readable descriptions instead of word distributions given by the topic model. To crystallize this idea, we first generate n-gram phrases, followed by predefined POS patterns (*cf.* Table 1) [12]. Then, we determine the most suitable phrase to represent a given topic by calculating their generated probability:

$$P(Phrase_m|T_k) = \frac{P(T_k \mid Phrase_m)P(Phrase_m)}{P(T_k)}. \quad (2)$$

where $P(T_k)$ can be ignored since it doesn't affect the ranking result, and $P(Phrase_m)$ is calculated using a background n-gram model [14]. On the other hand, $P(T_k|Phrase_m)$ is computed by an EM-like procedure [6]. To sum up, the phrase prior, $P(Phrase_m)$, is used to determine the weights of the phrase and the likelihood, $P(T_k|Phrase_m)$, is used to measure the relevance degree between a pair of topic and phrase. However, a phrase usually consists of only a few words, the corresponding likelihood score might be un-accurately estimated. With the alleviation of this deficiency as our motivation, we can explore effective query expansion techniques, such as the Rocchio's method [2], to enhance the original phrase representation with the pseudo-relevance technique. A special property in the task is that each candidate phrase is exactly occurred in some training documents, thus we can treat those documents, which contain the given phrase exactly, as the "pseudo-relevant" documents directly. Thus, the frequency count of each word in an updated phrase can be calculated by:

$$\begin{aligned}\hat{c}(w_i, phrase_m) &= c(w_i, phrase_m) \\ &\quad + \sum_d c(phrase_m, d)\cdot c(w_i, d).\end{aligned} \quad (3)$$

Consequently, each topic is expressed by a meaningful phrase which is friendlier for users to understand the physical meaning of the topic.

C. Concept Merge

Although, topic modeling can be used to dissect word usage cues and then generate a predefined numbers of topics, it may generate similar topics since it is an unsupervised framework. To handle the foreseeable problems, we hence define a similarity function to merge similar concepts based on KL-divergence measure [2, 9]:

No.	POS patterns
1	NN NN+
2	JJ+ NN+
3	NN+ JJ+ NN+
4	NN+ IN+ NN+
5	NN+ VB+ JJ+
6	NN+ VB+ (TO|IN)* VB+ CC* VB* NN+
7	NN+ JJ+ (VB+ (TO|IN)* DT* JJ* NN*)* NN+

Table1. POS patterns used in this paper, where NN is Noun, JJ is adjective, VB is verb, TO is to, IN is preposition or subordinating conjunction, CC is coordinating conjunction, DT is determiner, + is one or many, and * is zero or many. Please refer to Penn Treebank Project [10] for details listing of POS tags.

$$\begin{aligned}&Sim(T_k, T_{k'}) \approx Sim(Phrase_k, Phrase_{k'}) \\ &= -\frac{1}{2}\left[KL(Phrase_k \parallel Phrase_{k'}) + KL(Phrase_{k'} \parallel Phrase_k)\right].\end{aligned} \quad (4)$$

If the similarity score between a pair of phrases are higher than a threshold, these two topics will be merged.

D. Search

Traditional information retrieval task focuses on ranking a set of documents with respect to their own relevance degrees between a given query and each document. The retrieval results usually consist of all the documents. However, search in the IT ticket delivery environment needs to address both the precision and confidence for each document extracted by a query (topic). This search protocol is due to high penalty of human cost incurred by search errors. Although simple strategies are available, such as normalized query likelihood score with a predefined threshold, it is hard to determine a good threshold and the threshold will be sensitive to the total number of documents in the data set. To mitigate such problem, we define a score, independent to the number of documents in the data set, for each pair of document and topic:

$$S(d_j, T_k) = \frac{P(d_j \mid T_k)}{\sum_{k'=1}^{K} P(d_j \mid T_{k'})} \approx \frac{P(d_j \mid Phrase_k)}{\sum_{k'=1}^{K} P(d_j \mid Phrase_{k'})}. \quad (5)$$

The likelihood score can be further decomposed by [2, 14]:

$$\begin{aligned}P(d_j \mid Phrase_k) &= \prod_{w \in d_j} P(w \mid Phrase_k)^{c(w,d_j)} \\ &= \prod_{w \in d_j} \left[P_U(w \mid Phrase_k) + P_T(w \mid Phrase_k) + P_{BG}(w)\right]^{c(w,d_j)},\end{aligned} \quad (6)$$

where $P_U(w|Phrase_k)$ is the probability of word w occurring in k-th phrase, $P_T(w|Phrase_k)$ is computed by integrating both the word distribution of each topic $T_{k'}$ (i.e., $P(w|T_{k'})$) and the phrase likelihood score of each topic $P(T_{k'}|phrase_k)$, and then $P_{BG}(w)$ is determined by the background language model. By doing so, both literal matching (such as password reset vs. reset password) and concept matching (such as purchase pc vs. buy desktop) can be considered into the ranking score. If the score $S(D_j, T_k)$ is greater than a threshold, the document D_j is assigned to topic T_k. It is worthy to notice that the score nicely fits into the practical scenario that is the document may belong to multiple topics. As a result, a topic has its own document set, and an IR-like system can further be leveraged to re-rank these

#	Topic names	Tix count
1	user try to login	123
2	issue no longer exist confirm with the user	429
3	clear rmt cache	183
4	client mobile thin client	472
5	request for information	291
6	further assistance	355
7	desktop optimization program	109
8	issue detail	89
9	user agreed to close ticket	248

Table 2: topic name and distribution from AP resolution

documents. The ranking list and the representative phrase of each topic can be obtained finally.

IV. Experiments

A. Setup

We use two service desk ticket sets for our experiments. These tickets are collected from 2013 summer to 2014 spring, from a production system. One of the ticket sets in our experiments is related to mailbox problems and the other is related to Applications Portals (AP). The two sets represent two different use cases. The mailbox tickets are more specific, which relates all possible issues for the email system. On the contrary, the AP tickets, which related to many business applications, cover boarder spectrum. Since the mailbox system is one of the supported enterprise applications, the AP tickets could contain mailbox related tickets. All of the tickets are managed by an IPC system which contains both structured and unstructured fields. The agent who addresses the tickets manually labeled the tickets into the structured fields using the predefined categories. Ideally, he also documents problems and resolutions details in ticket description and resolution fields, respectively, via text. As expected, due to the complexity of the tickets, time pressure, etc., the structured data could be miss-labeled (i.e., the free text can be noisy). Each set of tickets have approximately 20k tickets. We use only description and resolution text for our experiments.

B. Subjective Evaluation

Table 2 demonstrates samples generated by the proposed framework from AP ticket resolution. In this experiment, the number of latent topics is set to 33. Each topic is represented by the phrase with best score. The representative phrase was then used as query to extracted related tickets. From human evaluation, most of the topics are very encouraging, except some maybe too board, e.g. #8 "issue detail". The ticket counts have been scrambled by a monotonic scaling function; they are not true ticket counts. Yet, the table clearly shows the ideas of topic distribution of the ticket data.

The search results are also evaluated manually. The following lists illustrate example ticket resolutions extracted from the forth topic, "client mobile thin client". These tickets are concise with minor grammatical errors and somewhat technical; yet, the tickets are much related to client mobile thin client topic.

- *advised user to place cmp request for reassign existing desktop laptop thin client mobile thin client*
- *advised that he needs to order a reassign existing desktop laptop thin client mobile thin client through AP*

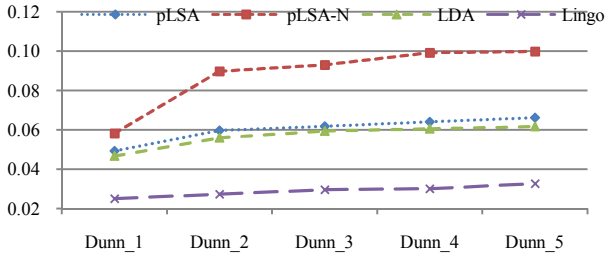

Figure 1. Dunn comparison for Lingo, LDA, pLSA and pLSA-N using AP tickets description. Notes: larger score is better.

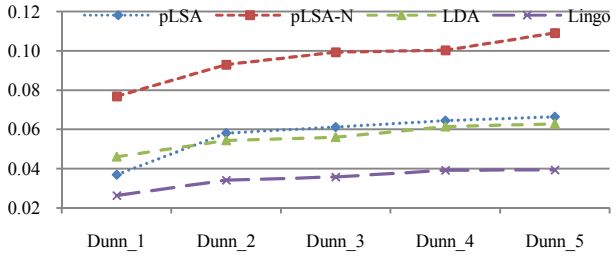

Figure 2. Dunn comparison for Lingo, LDA, pLSA and pLSA-N using AP tickets resolution. Notes: larger score is better.

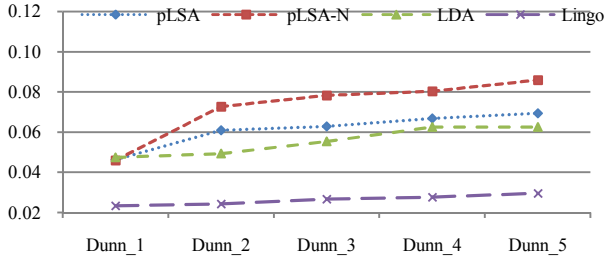

Figure 3. Dunn comparison for Lingo, LDA, pLSA and pLSA-N using Mailbox tickets description. Notes: larger score is better.

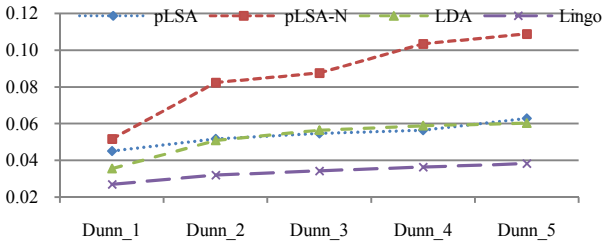

Figure 4. Dunn comparison for Lingo, LDA, pLSA and pLSA-N using Mailbox tickets resolution. Notes: larger score is better.

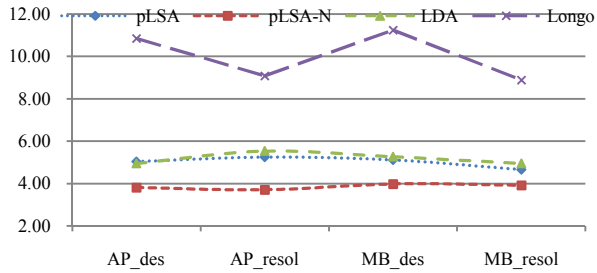

Figure 5. DBI comparison for Lingo, LDA, pLSA and pLSA-N using all data sets. Notes: smaller score is better.

- *steps taken last successful log in 05 09 2014 restart laptop advise customer to place a request through AP to accomplish this reassign existing desktop laptop thin client mobile thin client*
- *reached out to the user the application is installed on the users thick client as request in the cmp*
- *user restarted thin client and is now seeing his programs*

In general, these topics reveal potential pain points and the ticket counts illustrate the corresponding pain point distribution. In addition, the trend analysis can be achieved by analyzing tickets distribution for each topic as a time series. If there are any abnormal patterns, we can trigger a more detailed analysis for the tickets under these particular topics. Due to the space limitation, we can only present a set of experimental result.

C. Quantitative Evalution

Once we have encouraging initial results, we then compare the proposed framework to the Lingo approach [11]. Lingo is a well-practiced method that proposed a "description-comes-first" approach for topic clustering, and it has been packaged and is public accessible through Apache Carot-2 framework. Since Lingo supports the end-to-end process, we tried our best to sweep over the system parameters to ensure Lingo produces the best results for our comparison studies.

To support massive experiments with comparable quantities, we investigated the widely used clustering integrity matrices, Dunn index (Dunn) [4] and Davies-Bouldin index (DBI) [5] for evaluation. The score of Dunn index is calculated by the ratio of the minimum distance of inter-cluster and the maximum distance of inter-atom paired. It penalizes the worst scenario and is very sensitive to anomaly. To alleviate the worse scenario of the conventional Dunn index, which denoted as Dunn_1 in this paper, we also calculate Dunn index scores based on the 2^{nd}, 3^{rd}, 4^{th} and 5^{th} minimum and maximum distances. The modified Dunn index, called Dunn_2 to Dunn_5, can illustrate if Dunn indexes are skewed by some bad clusters. Another widely used clustering integrity matric, the DBI, calculates the averages distances difference for all sample pairs and clustering pairs. Figure 1 to 4 show Dunn 1-5 results for AP and mailbox tickets via description and resolution, respectively. Figure 5 shows DBI results. These figures clear demonstrate our approach consistently outperform Lingo in both modified Dunn and DBI matrices. In addition, pLSA is slightly better than LDA, except some experiments in Dunn_1. In addition to the basic results, we also explore using 3 phrases to represent a topic during search for topic clustering. This setup is running with pLSA preliminarily, denoted as pLSA-N in this paper. From figure 1 to 5, the pLSA-N clearly yields the best results.

V. Conclusion

We propose a simple yet effective probabilistic concept modeling framework. This approach extracts concepts from phrases, instead of words, and clusters documents using the concepts. This methodology has been applied to noisy texts like IT service desk tickets and the results have demonstrated its potential ability and capability when compared to existing method (i.e., Lingo). The concepts extracted from the data set demonstrate important topic addresses by the tickets. Search/clustering results illustrated which tickets are related to these topics. It can be used as trend analysis to lay out the strategies for future ticket prevention or tickets reduction. The framework is so effective, since it only takes a few minute to run the end-to-end pipeline for 30k tickets. In the further, we will continue on more advanced probabilistic topic modeling framework for ticket resolution recommendations.

Acknowledgment

The authors would to acknowledge IBM Global Technology Service colleagues: Dr. Nadeem Malik and Beth Rudden for providing domain knowledge, proof reading and data acquisition.

References

[1] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, 3, 2003.

[2] K. Y. Chen, S. H. Liu, B. Chen, E. E. Jan, H. M. Wang, W. L. Hsu, and H. H. Chen, "Leveraging Effective Query Modeling Techniques for Speech Recognition and Summarization," in *Proc. of EMNLP*, 2014.

[3] W. B. Croft and J. Lafferty (eds.), "Language Modeling for Information Retrieval," Kluwer International Series on Information Retrieval, Volume 13, Kluwer Academic Publishers, 2003.

[4] J. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *Journal of Cybernetics*, 3 (3), 1973.

[5] L. Davies and D. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-1 (2), 1979.

[6] T. Hofmann, "Probabilistic Latent Semantic Indexing," in *Proc. of SIGIR*, 1999.

[7] E. E. Jan, J. Ni, N. Ge, N. Ayachitula, and X. Zhang, "A Statistical Machine Learning Approach for Ticket Mining in IT Service Delivery," in *Proc. of IM*, 2013.

[8] T. K. Landauer, W. F. Peter, and L. Darrell, "An Introduction to Latent Semantic Analysis," *Discourse processes* 25.2–3, 1998.

[9] C. D. Manning, P. Raghavan and H. Schtze, *Introduction to Information Retrieval*, New York: Cambridge University Press, 2008.

[10] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The Penn Treebank: Annotating Predicate Argument Structure," in *Proc. of HLT*, 1994.

[11] S. Osinski, "Lingo: Search Results Clustering Algorithm Based On Singular Value Decomposition," in *Proc. of IIPWM*, 2004.

[12] R. Potharaju, N. Jain, and C. Nita-Rotaru, "Juggling the Jigsaw: Towards Automated Problem Inference from Network Trouble Tickets," in *Proc. of NSDI*, 2013.

[13] D. Rosu, W. Cheng, E. E. Jan, and N. Ayachitula, "Connecting the Dots in IT Service Delivery: From Operations Content to High-Level Business Insights," in *Proc. of SOLI*, 2012.

[14] C. X. Zhai and J. Lafferty, "A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval," in *Proc. of SIGIR*, 2001.