When are Latent Topics Useful for Text Mining?

Enriching Bag-of-Words Representations with Information Extraction in Thai News Articles

Nont Kanungsukkasem[1], Piyawat Chuangkrud[1,2], Pimpitcha Pitichotchokphokhin[1], Chaianun Damrongrat[2], and Teerapong Leelanupab[1,3](✉)

[1] Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok 10520, Thailand
{nont,piyawat,pimpitcha,teerapong}@it.kmitl.ac.th

[2] National Electronics and Computer Technology Center (NECTEC), Pathumthani 12120 Thailand
chaianun.damrongrat@nectec.or.th

[3] The University of Queensland, Brisbane, QLD 4072, Australia
t.leelanupab@uq.edu.au

Abstract. The Bag-of-Words (BOW) model is simple but one of the successful representations of text documents. This model, however, suffers from the sparse matrix, in which most of the elements are zero. Topic modeling is an unsupervised learning method that can represent text documents in a low-dimensional space. Latent Dirichlet Allocation (LDA) is a topic modeling technique used for topic extraction and data exploration, with interpretable output. This paper presents a thorough study of potential benefits of applying LDA, as a feature extraction, to topic discovery and document classification in Thai news articles, comparing with TF–IDF and Word2Vec. We also studied how much of the top Thai terms extracted from LDA with the different numbers of topics can be interpretable and meaningful, and can be a representative of the corpus. Besides, a set of Topic Coherence measures were included in our study to estimate the degree of semantic similarity of extracted topics. To compare the performance and optimization time of classification of features from the different feature extraction methods, various types of classifiers, e.g., Logistic Regression, Random Forest, XGBoosting, etc., were experimented.

Keywords: Topic Modeling · Latent Dirichlet Allocation · Word Embedding · Bag of Words · Text Mining · Thai News

1 Introduction

In a Bag-of-Words (BoW) model, a text document is represented as a distinct vector of *weights* of tokens, indexed words or terms in a vocabulary. The weights from a term weighting indicate the importance of terms in a document and/or their discriminative power in differentiating one document from the others on

N. T. Nguyen et al. (Eds.): ACIIDS 2023, CCIS 1863, pp. 205-219, 2023.
https://doi.org/10.1007/978-3-031-42430-4_17

specific tasks, though it lacks perception of word morphology, grammar and word order. Examples of term weighting are raw or normalized term frequency (TF), variants of TF-Inverse Document Frequency (TF–IDF) and BM25 weighting. Generally used in natural language processing (NLP), information retrieval (IR) and machine learning (ML), the BoW model has several good reasons owing to its simplicity and robustness. Previous studies showed that simple systems, e.g., in IR and ML, using large amount of data could outperform complex ones using fewer data [8]. As a trade-off for performance, BoW-based systems sacrificed their computational cost due to high dimensional feature vectors regarding a large vocabulary. However, BoW does not consider similarity between words and co-occurrence statistics between words.

Word embedding is a dense continuous word representation, capable of capturing the syntactic and semantic relationship of words. Focusing on the sequential combination of words, word embedding models assume that the appearance of each word is only related with a limited set of words before it. Commonly available and notable pre-trained word embeddings include Word2Vec [11]. In this paper, we utilize Word2Vec as a representative approach from word embedding to reduce a document representation from based on words to based on sentences in a document.

Towards dimensionality reduction and semantic information extraction, topic modeling is one of the unsupervised learning techniques for document representation. Independent on any language, topic modeling can reduce a noisy BoW to a more compact representation based on topics. Regarded as the state-of-the-art topic modeling method, Latent Dirichlet Allocation (LDA) [3] showed better performance than Non-negative Matrix Factorization (NMF), Latent Semantic Analysis (LSA) and probabilistic LSA (pLSA) [16].

1.1 Goals of the Paper

The main goal of this paper is to conduct a comprehensive study of potential advantages of applying latent topics, as extracted features, to text mining tasks in Thai news articles. In general, Thai language is considered more complex to mine than others. This is due to the lack of word boundary defined in a Thai sentence, introducing ambiguity in word tokenization. Topic modeling is a language-independent technique that can reduce such complexity. However, there have been a few studies of topic modeling in Thai corpora [16]. This paper aims to answer the following research questions by conducting two sets of experiments regarding two text mining tasks (i.e., topic discovery and text classification):

Q1: How does LDA perform in discovering a set of k topics, represented by *top-ranked terms*? Are the top-ranked terms for each topic meaningful and interpretable, especially for the Thai language?

Q2: How can we define the number of k topics on modeling? Does topic coherence provide a rough estimate of the number of topics discovered by LDA?

Q3: Other than the benefit of meaningful and interpretable features from LDA in Q1, how much do the performance and computational trade-off of TF–IDF, LDA with three different numbers of k topics and Word2Vec gain or lose in *text classification*

1.2 Previous Work

Li *et al.* [9] proposed a new model for clustering short English texts from academic abstracts, represented by paragraph features of Word2Vec and topic features of LDA as well as their unique embeddings derived from the combination of the two features. They compared the performance regarding clustering performance with a traditional TF-IDF BoW model. Inspiring the work of Li *et al.*, a hybrid approach of Wang *et al.* [18] used both Word2Vec and LDA as document features. By ad-hoc varying the number of topics, Wang *et al.* however only studied on the aspects of topic distribution over terms and distance between discovered topics. Instead of using Word2Vec. Asawaroengchai *et al.* [2] added the contextual relationships among words to all topics in a semantic space by using N-gram as input to LDA. In comparison with a traditional LDA, their Topic N-grams model was evaluated on a BEST2010 Thai corpus. Nararatwong *et al.* [14] simply improved topic extraction of LDA in Thai tweets by adding a refined stop-word list as a text pre-processing step.

2 Experimental Design

2.1 Data Preprocessing

We conducted the experiment on Thai news articles from BangkokBiz news website[1], published in separate categories. We collected 30,092 news articles, excluding their headline, from seven main categories, i.e., Politics, Finance, World, Economic, Lifestyle[2], Business, and Royal from April 11, 2019 to March 30, 2020 by using Beautiful Soup library. The numbers of documents out of 30,092 in each category are 8,567, 7,379, 5,485, 3,853, 3,577, 864 and 367, respectively.

PyThaiNLP library for Thai text processing provides modules to support all four steps in data pre-processing, i.e., word tokenization, stopword removal, stemming and noise removal. The library provides many tokenization algorithms (i.e., newmm (default), longest, deepcut, attacut, icu and ulmfit) to choose. However, Chormai *et al.* [7] showed that deepcut was better than the others in term of segmentation quality but worse in term of computational time. We also confirm the Chormai's findings in our pilot study that newmm is inferior to deepcut. For example, “หัวเว่ย”, which is transliterated from “Huawei”, was erroneously tokenized to two separated tokens, “หัว” for “Hua” and “เว่ย” for “Wei”, by newmm, but was correctly tokenized to “หัวเว่ย’ by deepcut. Accordingly, we chose to use deepcut exploiting the convolutional neural network to tokenize our dataset after removing the characters that were not letters or vowels. Then, low-frequency tokens appearing less than five times were filtered out. Afterwards, we filtered out all function words in Thai and English by using two stopword lists as provided by PythaiNLP and Natural Language Toolkit (NLTK), respectively. The preprocessing of 30,092 articles resulted in a total of 5,898,527 tokens, approximately

[1] https://www.bangkokbiznews.com

[2] “Lifestyle” category includes contents from other subcategories, e.g., health and sport.

196 tokens on average per article. Out of these tokens, 29,537 were unique. The preprocessed articles were then randomly splitted into 70% for training and 30% for testing which is 21,064 documents with 29,220 unique tokens for training and 9,028 documents with 26,565 unique tokens for testing.

2.2 Feature Extraction

To answer Q3, we selected TF-IDF, LDA and Word2Vec for comparison. They were applied to extracting features from the preprocessed articles. We chose to use Scikit-learn to extract the articles into 29,220 TF–IDF (BoW) features. Then, we consider the final results from these features as a baseline for Q3. Accordingly, we chose to use Gensim that features both LDA and topic coherence. Practically, the proper numbers of topics and iterations have to be investigated by a preliminary experiment.

To answer Q1, the top-ranked terms of k topics must be interpreted to compare with the seven collected categories of the news articles to show whether the latent topics from LDA can represent all of the categories. Accordingly, we started our experiment with seven as the number of topics for LDA (LDA7), resulting in seven features for training a model. However, setting the number of topics to be the same as the number of categories of a corpus is not practical with other datasets as they are not pre-categorized. Besides, LDA is an unsupervised algorithm to find latent topics, by which we in practice do not know the actual number of topics. Then, we determined the number of topics using the topic coherence scores of the results from LDA with different numbers of topics ranging from 1 to 50. However, as LDA is a generative probabilistic model, the estimation is not always the same. Accordingly, for each number of topics, we experimented ten times to get its average coherence score.

Furthermore, we experimented all four topic coherence measures provided by gensim, i.e., UMass, UCI, NPMI and CV to answer Q2. When the number of topics, as suggested by the topic coherence, was not equal to seven or not the same as the number of seven main categories that we had collected, we would get two sets of the top-ranked terms from LDA. Otherwise, there would be only one set of the top-ranked terms to be further used for answering Q1 and Q2. Also, LDA with two different numbers of topics were then used to extract features for the next step to answer Q3. Gensim also features Word2Vec algorithm including both Skip-gram and Continuous Bag-of-Words (CBOW) models. As Mikolov *et al.* [11] suggested Skip-gram provided a better semantic accuracy than CBOW, we therefore applied Skip-gram as a training algorithm to Word2Vec and used default settings for the other parameters in our study.

We further set the dimensionality of the word vectors to 300 and set the context (window) size to five according to Mikolov *et al.* [12]. As the number of features extracted from Word2Vec is 300 (W2V), we also set the number of topics in LDA to 300 (LDA300) to get the same number of features from W2V in order to set a fair comparison between them. Accordingly, there were five sets of features for our comparative experiment.

2.3 Modeling and Evaluation

To answer Q3, we measure the performance and computational trade-off when applying different types of features (i.e., TF–IDF, LDA, Word2Vec) to a downstream task (e.g., multi-class text classification.) We therefore studied on various machine learning algorithms to classify Thai news articles into seven classes, labeled by the actual categories of our dataset. These algorithms included Logistic Regression (LR), Multilayer Perceptron (MLP), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Adaptive Boosting (ADAB), GradientBoosting (GBM) [4] and XGBoosting (XGB) [6].

We performed the model optimization by tuning hyperparameters using GridSearchCV with k=5, to cross-validate each classifier with its set of permuted parameters to control its learning process. The best parameters of each classifier were maintained to fit the model on a training set, previously split by a simple hold-out method. Each trained model was subsequently validated on the remaining test set. All experimental runs were conducted on Google Cloud Platform by running on virtual machines with the specifications; zone: asia-southeast1-b, machine type: n2-custom (8 vCPU, 32 GB memory), boot disk: balanced persistant disk (50 GB) and OS: Ubuntu 18.04 LTS.

To evaluate the performance of a classifier with different sets of features, we employed two evaluation metrics; accuracy and macro F1. Those metrics are suitable for multi-class classification problem, and especially when we have an imbalanced dataset but all classes are equally important. Computational time for tuning hyperparameter by GridsearchCV was also reported to compare the time spent on fitting and tuning models by features from different extraction methods. Lastly, the trade-off between performance and time was computed by the fraction of the performance gain over Time Loss (TL). When considering the performance gain by Accuracy Gain (AG), we call it Accuracy-to-Time (AT) ratio defined as:

$$\text{AT-ratio} = \frac{\text{AG} + \epsilon}{\text{TL} + \epsilon} \tag{1}$$

where ϵ is a very small constant that is added to the denominator to avoid problems of division-by-zero and added to the numerator to avoid misinterpretation when the numerator is equal to 0. For example, when accuracy values of two experiments are the same number as the minimum accuracy of all experimental runs but with different time losses, the one with the lower time loss should be considered as a better one. Provided epsilon was not added to the numerator, two experiments would be considered the same because both of them would be equal to zero. Besides, the addition to both numerator and denominator also gives us the number 1 as a baseline, which is the number of the ratio when an experiment performs the worst but spends the least computational time, instead of 0/0. The epsilon was set to 0.001 in our experiment.

AG is calculated from the Accuracy metric (*acc*) and the minimum of Accuracy of all experimental runs[3]. The AG can be formalized as follows:

[3] As Accuracy is in percentage, we do not need any normalization like TL.

$$\mathrm{AG} = acc - \min(acc) \tag{2}$$

TL is the difference between a computational time (t) and the minimum computational time of all experimental runs scaled by Min-Max normalization.

$$\mathrm{TL} = \frac{t - \min(t)}{\max(t) - \min(t)} \tag{3}$$

When we consider the performance gain by F1 Gain (FG), we call it F1-to-Time (FT) ratio. It can be derived by simply replacing AG with FG in the AT ratio where FG can be calculated by the following equation from the macro-F1:

$$\mathrm{FG} = F1 - \min(F1) \tag{4}$$

3 Results and Discussions

3.1 Q1 and Q2

In each of the seven topics extracted by LDA7, we retrieve the top ten terms and present them in Table 1[4]. By interpreting all terms together, we can assign a label to each topic. Labels are shown after the topic numbers in parenthesis, such as Finance, Economy, Politics, and so on. Ideally, these labels should be aligned with the categories we collected from BangkokBiz (see Section 2.1.) Some topics are duplicate. For example, topic 1 and 3 are both labeled with Finance, and topic 4 and 5 are also labeled with Politics. In contrast, some categories are missing and cannot be discovered by LDA7, i.e., Royal, Business and Lifestyle. However, for the "Lifestyle" category, it is instead actually labeled with its subcategory, "Health" and "Disaster". Topic 3 can be interpreted and assigned with three labels, i.e., Finance, Economy and World. In our view, the imbalance of our data might be one of the reasons of lacking the categories, "Royal" and "Business".

Table 1. Seven topics extracted by LDA7

Topic 1 (Finance)	บาท ล้าน เงิน ปี ทุน หุ้น ลด ค่า บริษัท ราคา
Topic 2 (Economy)	ประเทศ ปี งาน ไทย ทำ ธุรกิจ พัฒนา สินค้า สร้าง โลก
Topic 3 (Finance/Economy/World)	ตัว สหรัฐ ราคา ตลาด จีน ลด น้ำมัน ดอลลาร์ จุด เงิน
Topic 4 (Politics)	คน พรรค ทำ รัฐบาล เรื่อง เมือง ประชาชน ตัว เลือกตั้ง นายก
Topic 5 (Politics)	รัฐมนตรี ประชุม ข้อ คณะ พิจารณา เรื่อง กฎหมาย คดี งาน ประธาน
Topic 6 (Disaster)	น้ำ พื้นที่ ทำ เรียน ศึกษา เด็ก งาน ภัย รถ ปี
Topic 7 (Health)	โรค คน เชื้อ ติด ไวรัส ป่วย ระบาด ประเทศ บิน เดินทาง

As it is not practical to know the number of topics, we experimented on LDA with different numbers of topics ranging from 1 to 50 to find the potentially

[4] We provide a hyperlink for each Thai word leading to its meaning in English.

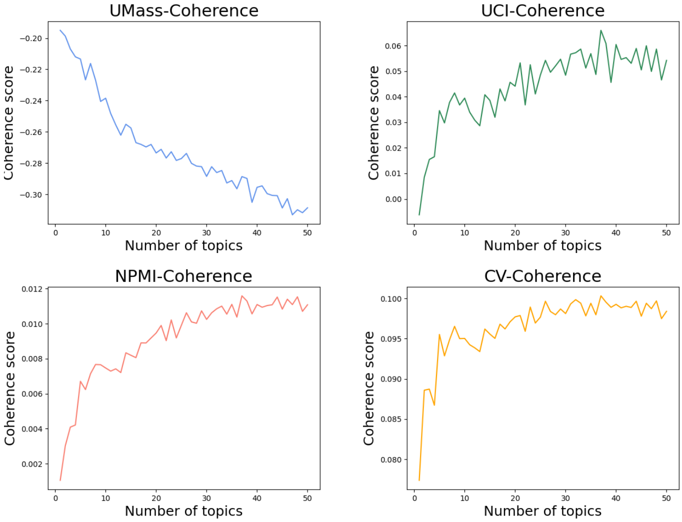

Fig. 1. The average coherence scores of LDA as evaluated by four different metrics i.e., UMASS, UCI, NPMI and CV, respectively.

optimal number of topics. The result plotted in Fig. 1 shows that topic coherence scores from UCI, NPMI and CV have the same elbow at 37 but 47 from UMass. According to the majority voting among studied topic coherence metrics, we chose 37 to be the potentially optimal number of topics for fitting LDA on our corpus. We later name this method LDA37 for feature extraction.

Again, we retrieved the top ten terms in each of the 37 topics extracted by LDA37. Table 2[4] demonstrates examples of the top ten terms in 13 out of 37 topics[5]. As we can see, they cover all the seven categories with a lot of subcategories. Even though there are only 367 documents in the "Royal" category which is only 1.2% of the total document in our corpus, LDA37 can extract the topic, "Royal", which is interpreted from top ten terms in topic 6. An example of all seven categories extracted from LDA37 is topic 1, 2, 3, 6, 8, 13 and 14 that can be interpreted easily to be the same as Finance, Lifestyle, Economy, Royal, Business, World and Politics categories, respectively.

Some topics from LDA37 are more specific than those from LDA7. For instance, topic 4 is about the protests and demonstrations in Hong Kong which happened around the time we collected the data, and topic 12 is specifically about COVID. Topic 12 is separated from topic 10 which is about "Health" unlike topic from LDA7 that has only one "Health" topic. Even though some topics are duplicate in broad interpretation, they are still different when we did deeper interpretation. However, a few topics are difficult, but possible, to be interpreted deeper by human.

[5] All 37 topics and 300 topics can be viewd via the provided link attached to this footnote.

Table 2. Top ten terms of thirteen example topics from total 37 topics extracted by LDA37. Identification of each topic is denoted by "T", followed by its identifier number.

T01 (Finance)	T02 (Lifestyle/ Travel)	T03 (Economic)	T04 (World/ Political)	T05 (Agriculture/ Farming)	T06 (Royal)	T07 (Econo-/ Finance)	T08 (Business)	T09 (World/ Lifestyle)	T10 (Health)	T11 (Development)	T12 (Health/ COVID-19)	T13 (World)
ล้าน	บิน	ปี	ฮ่องกง	น้ำ	ตำแหน่ง	สหรัฐ	ธุรกิจ	ร้อย	โรค	งาน	เชื้อ	ประเทศ
หุ้น	ท่องเที่ยว	เศรษฐกิจ	ตำรวจ	เกษตรกร	ราชการ	เดือน	บริการ	ปี	ป่วย	พัฒนา	ไวรัส	ปี
ปี	เที่ยว	ตัว	ทหาร	ข้าว	พระราชทาน	ตัว	ลูกค้า	ระบุ	ยา	ระบบ	ระบาด	ญี่ปุ่น
บาท	เครื่อง	ลด	ประท้วง	สัตว์	เสด็จ	จุด	ทำ	อันดับ	แพทย์	โครงการ	คน	ล้าน
ทุน	เดินทาง	กระทบ	ชุมนุม	สาร	ประกาศ	อังกฤษ	ตลาด	เมือง	อาการ	สร้าง	โรค	โลก
บริษัท	โดยสาร	ไทย	เจ้าหน้าที่	เกษตร	พศ	ระดับ	ออนไลน์	โลก	รักษา	ประเทศ	ติด	บริษัท
ราคา	สาย	ประเทศ	คน	ปลูก	ดำรง	ดัชนี	ดิจิทัล	สำรวจ	พยาบาล	ระดับ	โควิด	เอเชีย
ขาย	คน	ทุน	กอง	ผลิต	พระราชพิธี	การณ์	สร้าง	คน	สุขภาพ	นโยบาย	แพร่	อินเดีย
ซื้อ	ไทย	คาด	เหตุการณ์	พืช	king *	ร่วง	เติบโต	ตัวอย่าง	กัญชา	ดำเนิน	ประเทศ	เวียดนาม
กำไร	เส้นทาง	โลก	ทัพ	ปริมาณ	แต่งตั้ง	เกี่ยว	ยอด	อายุ	ติด	ทำ	สถานการณ์	สิงคโปร์

*พระบาทสมเด็จพระเจ้าอยู่หัว

In addition to LDA7 and LDA37, we performed LDA with 300 as the number of topics (LDA300) in order to get the same number of features or feature vector length as that of Word2Vec (W2V). However, as it is not possible to show all 300 topics, we provide only some important aspects from the results of LDA300 to compare them with those of LDA7 and LDA37. The results from LDA300[5] can cover all seven categories. However, as 300 is a lot higher than seven, the actual number of categories, and set without any theory support, many of the latent topics from LDA300 are too ambiguous to be interpreted and many of them can be interpreted to be the same topics. Additionally, nine topics have the exact same top ten terms with the same order.

In summary, the top-ranked terms of seven topics from LDA7 are the easiest to be interpreted and very meaningful, but cannot represent all seven categories of our corpus. Furthermore, the number of topics cannot practically known beforehand. So, we set a preliminary experiment on LDA with the different numbers of topics, compared their topic coherence scores and got 37 as the potentially optimal number of topics for our corpus. The top-ranked terms of 37 topics from LDA37 are interpretable though a bit difficult for a few topics, and meaningful enough to give the rough idea of the context possibly from the topics. They cover all seven categories and give us a lot of latent topics that is comparable to subcategories of our corpus. Accordingly, Q2 can be answered that we can define the number of topics by experimenting with various numbers of topics and we can use topic coherence scores to get a rough estimate of the number of topics. Besides, LDA with 300 was additionally performed. The top-ranked terms of 300 topics from LDA300 are difficult, if impossible, to be interpreted and some of them are not meaningful at all. As a result, we can answer Q1 that LDA with the potentially optimal number of topics gives us the best set of latent topics represented by top-ranked terms that is interpretable and meaningful.

3.2 Q3

Table 3 shows the performance (i.e., Accuracy and macro F1) and computational time of classification algorithms using five comparative sets of features. In Ta-

Table 3. Performance and computation time of each classification algorithm with different feature extraction methods.

	Accuracy (percent)					macro F1 (percent)					Computational time (seconds)				
	BoW	LDA7	LDA37	LDA300	W2V	BoW	LDA7	LDA37	LDA300	W2V	BoW	LDA7	LDA37	LDA300	W2V
LR	87.96	67.79	81.40	85.16	**88.37**	84.01	46.16	73.10	80.43	**84.49**	971	**30**	35	60	767
SVM	87.31	73.14	84.49	86.83	**88.72**	82.88	60.04	78.92	82.68	**84.47**	1226	**93**	137	718	607
MLP	88.21	72.15	83.27	86.15	**88.24**	**83.82**	58.67	77.15	81.02	83.74	1300	**53**	81	169	520
DT	74.52	70.46	76.72	74.49	**75.14**	**66.91**	55.78	66.45	67.82	65.77	160	**6**	20	91	300
RF	83.93	74.03	83.62	84.38	**86.66**	75.72	62.27	76.23	77.65	**81.54**	422	**108**	185	367	944
ADB	85.20	72.77	83.98	85.26	**87.11**	79.88	61.39	78.12	80.63	**82.27**	10493	**271**	854	4208	10596
XGB	87.87	74.90	84.37	87.13	**88.48**	83.45	63.23	78.52	82.93	**84.70**	16238	**3295**	7012	22109	46961
GBM	86.22	74.50	83.54	85.06	**87.43**	80.61	61.64	75.78	76.89	**81.78**	4839	**631**	1403	4576	29360

Note: The values in **bold** show the best among feature extraction methods and in underline show the best among learning algorithms.

ble 4, we calculate and report the trade-off between performance gain and time loss, shown by AT and FT ratios.

When considering among LDAs with the different numbers of topics (i.e., LDA7, LDA37 and LDA300), the features from LDA7 classified by DT (LDA7-DT) spent the least time for optimization. Additionally, when considering only features from LDA7, LDA7-DT was also the best in term of trading off according to both ratios. However, LDA7-DT performed the worst with 70.46% accuracy and 55.78% macro F1 among different feature extraction methods and different algorithms. Besides, DT performed the worst with four sets of features and the second worst with a set of features.

Among LDAs, the XGB classifier trained with LDA300 features (simply denoted as LDA300-XGB) showed the best performance with 87.13% accuracy and 82.93% macro F1 but the most computational time, 22108s. However, considering only features from LDA300, LDA300-LR gave the best results in terms of trading-off according to both AT and FT ratios. Even though almost all of the algorithms performed the best with the features from LDA300, they spent the most computational time in comparison with the other LDAs. Accordingly, when considering with trade-off, the set of features from LDA37 was the best for all classification according to AT ratio and the best for 5 algorithms and the second for 3 algorithms according to FT ratio. Besides, LDA37-LR was the best according to both AT and FT ratios.

Among all feature extraction methods in our experiment, LDA-DT was still considered the best in term of computational time but the worst in term of performance. However, even the performance of XGB-LDA300 was the best in LDA-based runs, it still performed worse than many of those based on BoW and W2V. Considering accuracy, the features from W2V classified by SVM (W2V-SVM) showed the best performance at 88.72% accuracy with only 60s optimization time. In contrast, considering macro F1, the features from W2V classified by XGB (W2V-XGB) showed the best performance with 84.70% micro F1 but with the longest optimization time at 46961s. When considering trade-off, the best among all feature extraction methods were the same as the best among LDAs. However, when we considered only the results with more than 80% in both accuracy and macro F1, LDA300-LR was the best in term of trade-off with 81.48 AT ratio and 160.30 FT ratio. Besides, comparing W2V-SVM with

Table 4. Accuracy-to-Time (AT) and F1-to-Time Gain (FT) ratios of each classification algorithm with different feature extraction methods.

	AT ratio					FT ratio				
	BoW	LDA7	LDA37	LDA300	W2V	BoW	LDA7	LDA37	LDA300	W2V
LR	9.41	0.67	**84.65**	81.48	12.02	17.62	0.67	**166.96**	160.30	22.34
SVM	7.27	19.08	**44.30**	11.85	15.25	13.65	48.94	**86.65**	22.66	27.86
MLP	7.19	22.26	**60.06**	41.40	17.21	13.23	62.94	**119.86**	78.40	31.55
DT	15.94	27.70	**70.07**	24.29	10.28	48.65	97.20	**158.21**	77.74	27.19
RF	16.48	19.95	**33.16**	19.23	9.04	30.09	51.00	**62.80**	36.40	16.92
ADAB	0.78	7.64	**8.55**	1.94	0.86	1.51	**23.06**	16.82	3.82	1.60
XGB	0.58	1.01	**1.11**	0.41	0.21	1.08	**2.42**	2.16	0.78	0.39
GBM	1.78	4.76	**5.15**	1.77	0.32	3.32	**10.89**	9.66	3.14	0.57

Note: The values in **bold** show the best among feature extraction methods and in underline show the best among learning algorithms.

LDA300-LR, the performance between these two were not much different but the computational time of W2V-SVM was slightly tenfold greater than that of LDA300-LR. Accordingly, LDA300-LR seemed be the best choice according to our cross comparison of performance and computational time from five sets of features classified by eight algorithms. It took not much computational time and gave only a bit lower performance than the best one and got the highest ratios among the other features with over 80% in both accuracy and macro F1.

In summary, on average, W2V was the best in term of performance but the worst in term of optimization time and the second worst in term of trade-off and LDA 7 was the best in term of optimization time but the worst in term of performance and in the middle among all features extraction methods in term of trade-off. Even though LDA300 was in the middle in both performance and optimization time, its ratios did not the show the best trade-off but LDA37's ratios did. However, when specifically considering only the performance with over 80% in both accuracy and macro F1, LDA300-LR performed fairly good with not much time and got the highest score from both ratios.

4 Document Representations

4.1 Term Frequency-Inverse Document Frequency (*tf-idf*)

tf-idf [10] is a traditional method for term weighting in a BoW model. *tf* quantifies how important a term t is in a document, and *idf* quantifies how common the term t is among the corpus. Then, *tf-idf* is simply the product of *tf* and *idf*. There are many variant of *tf-idf*, especially for the *idf* component.

idf_t uses logarithm to reduce the effect of a fraction of the total number of documents (N) over the number of documents that the term t occurs (df_t). Both numerator and denominator are added by 1 to avoid a division-by-zero problem. This experiment used *tf-idf* function in Scikit-learn with its default parameters. Therefore, the constant 1 is added more to the *idf* after applying logarithm to avoid $idf = 0$ due to the ignorance of the term that appears in all documents.

$$idf_t = \log_e\left(\frac{N+1}{df_t+1}\right) + 1 \tag{5}$$

4.2 LDA (Latent Dirichlet Allocation)

LDA is a type of statistical model for discovering latent topics from a collection of documents, by inferring the relationship between terms, documents and topics in a corpus. Blei *et al.* [3] introduced LDA as an unsupervised topic model. It has become one of the most widely used topic models.

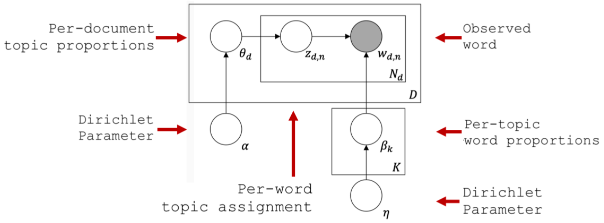

Fig. 2. The graphical model of LDA

The LDA model has the assumption that each of the n-th observed word $w_{d,n}$ in document d is generated by the other unobserved variables as shown in Fig. 2. In this representation, β_k denotes the word distribution of topic k, θ_d denotes the topic distribution of document d, and $z_{d,n}$ denotes the topic number of word n in document d. Each word is assigned as an index in the vocabulary, $w_{d,n} \in \{1, ..., V\}$ when a corpus of D documents contains V vocabulary words, and document d consists of N_d words, $(w_{d,1}, ..., w_{d,N_d})$. Additionally, η and α are Dirichlet parameters for β_k and θ_d, respectively. LDA also relaxes its assumptions to: *i)* the order of documents are not important. *ii)* the order of terms are not important. *iii)* the numbers of topics, K, is known and constant.

Given all words in all documents, the value of the unobserved variables in the model can be estimated by computing the posterior distribution to get the final results from LDA: β_k, each of which represents a latent topic $k \in \{1, ..., K\}$, and θ_d, each of which represents a proportion of topics per document calculated from $z_{d,n}$. Then, θ_d may be used as a representative or features of the document. The approximation of the posterior can be computed by inference algorithms, e.g., Gibbs sampling and Variational Bayes, to infer the variables.

4.3 Word2Vec

In NLP tasks, a BoW model shows only how frequent a word occurs in a document, but does not show similarity between words. Afterwards, Mikolov *et al.* [11] introduced two unsupervised models, Continuous Bag-of-Words (CBOW) model and Skip-gram models, both of which are architectures for computing representations of words in a continuous vector form by using neural networks. The goal of the architectures is the weights of hidden layer that need to be trained by backpropagation from a large dataset. Then, the weights become the continuous vector representations of words, called word embedding. The number of dimensionality used to represent each word (aka. the number of nodes in the hidden layer of the neural network) can be any number. The larger dimensionality values, the more fine-grained relationships can be captured. However, a lower dimensionality may capture more general features of words whereas a higher dimensionality may overfit to specific contexts. CBOW is a model architecture with the fake task to predict a middle word based on its surrounding words, but Skip-gram is a model architecture with the reverse fake task of CBOW, predicting the surrounding words based on a given word. In fact, the predictions from Skip-gram are not its objective but word representations that are useful for predicting the surrounding words. So, given a training data with T words, the objective of Skip-gram model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq j\neq 0}\log p(w_{t+j}|w_t) \tag{6}$$

where c is the context (window) size of surrounding words from the center word w_t. In theory, the probability in Equation 6 can be computed by a softmax function. However, when the size of the vocabulary is large, it is intractable to compute. Then, the approximation by a hierarchical softmax or negative sampling comes to make it feasible to compute [12]. The negative sampling is used by default in gensim with 5 noise words

4.4 Topic Coherence

Topic Coherence is an evaluation metric for topic modeling. To assess overall topics' interpretability, it measures the degree of semantic similarity between high scoring words in each topic. Topic Coherence can also be used to optimize the number of topics of topic models, which is generally needed to be specified by human topic ranking. Although there are many topic coherence measures, our experiment calculated topic coherence by functions in Gensim which cover 4 models, i.e., UCI, NPMI, UMass, and CV.

For UCI, topic coherence is quantified by calculating the pointwise mutual information (PMI) of each word pair from N top words inferring a topic (see in Eq. 7.) Each probability in PMI can be estimated from any external corpus as formalized in Eq. 8. Newman *et al.* [15] suggested that UCI achieved the best result when the external corpus was the entire Wikipedia articles.

$$C_{\text{UCI}} = \frac{2}{N\cdot(N-1)}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\text{PMI}(w_i, w_j) \tag{7}$$

$$\text{PMI}(w_i, w_j) = \log\frac{p(w_i, w_j)+\epsilon}{p(w_i)p(w_j)} \tag{8}$$

However, Aletras and Stevenson [1] showed that the UCI coherence performed better with normalized PMI (NPMI) as purposed by Bouma [5]. When the PMI in the UCI coherence is replaced by the NPMI, Eq. 9, the *modified* UCI coherence is then called NPMI coherence.

$$\mathrm{NPMI}(w_i, w_j) = \frac{\mathrm{PMI}(w_i, w_j)}{-\log(p(w_i, w_j) + \epsilon)} \tag{9}$$

UMass coherence [13] is also based on co-occurrences of word pairs. However, instead of using the product of probabilities of two words as the denominator just as in PMI, UMass coherence uses the probability of one word (see Eq. 10.)

$$C_{\mathrm{UMass}} = \frac{2}{N \cdot (N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \tag{10}$$

CV coherence was proposed by Röder *et al.* [17] and described in a systematic framework of coherence measures that combines the indirect cosine similarity with the NPMI and the boolean sliding window.

5 Conclusion

In this paper, we focused on the comparison of performance, computational time and their trade-off of classification when the input features were extracted from different methods, TF–IDF (BoW), LDA, Skip-Gram Word2Vec (W2V), which gave the different numbers of features (Q3). However, the number of topics from LDA, which was the number of input features for classification, needed to be calculated (Q2). So, we studied more on LDA about representation of Thai categories by top ten terms extracted by LDA whether they could be interpretable and meaningfulness. (Q1).

The results showed that LDA7 could discover topics with the top-ranked terms that were easy to be interpreted. However, such discovered topics could not represent all the categories in our corpus. Besides, setting the number by this way in practice is unfeasible as we do not know the number of topics in advance. In comparison, the top-ranked terms from LDA37, of which the number of topics was estimated by topic coherence score, could represent all categories of our corpus including many subcategories (Q1 and Q2).

For a fair comparison with Word2Vec having 300 features, we compared the results of LDA300 in a classification task produced by several learning algorithms with five sets of features. In our view, LDA300 with logistic regression seemed to be a pretty good choice when we considered performance, computational time, AT ratio and FT ratio. When we concerned about performance the most, W2V was the best choice to choose but had a trade-off for a lot longer optimization time. Comparatively, when we concerned about optimization time the most, LDA7 was the best choice to choose but demanded a trade-off for the worst performance. However, in our view, if we had to pick one set of features without considering a classification algorithm, we would pick the features from LDA with its potentially optimal number of topics (LDA37 in our experiment.) This selection was because the features was interpretable, could represent the corpus well and got the best trade-off for all classification algorithms according to the AT ratio, and received the best for five algorithms and the second for three algorithms according to the FT ratio.

Acknowledgements This work was supported by KMITL Research Fund under Research Seed Grant for New Lecturer with grant number: KREF186507.

References

1. Aletras, N., Stevenson, M.: Evaluating topic coherence using distributional semantics. In: IWCS 2013 (2013)
2. Asawaroengchai, C., Chaisangmongkon, W., Laowattana, D.: Probabilistic learning models for topic extraction in thai language. In: 2018 5th International Conference on Business and Industrial Research (2018)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan) (2003)
4. Bonaccorso, G.: Machine learning algorithms (2017)
5. Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of GSCL. vol. 30 (2009)
6. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al.: Xgboost: extreme gradient boosting. R package version 0.4-2 **1**(4) (2015)
7. Chormai, P., Prasertsom, P., Rutherford, A.: Attacut: A fast and accurate neural thai word segmenter (2019)
8. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) NeurIPS. vol. 30 (2017)
9. Li, C., Lu, Y., Wu, J., Zhang, Y., Xia, Z., Wang, T., Yu, D., Chen, X., Liu, P., Guo, J.: Lda meets word2vec: A novel model for academic abstract clustering. WWW '18 (2018)
10. Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development **1**(4) (1957)
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 26 (2013)
13. Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: EMNLP (2011)
14. Nararatwong, R., Legaspi, R., Cooharojananone, N., Okada, H., Maruyama, H.: Solving the difficult problem of topic extraction in thai tweets. Journal of Telecommunication, Electronic and Computer Engineering **8**(6) (2016)
15. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: NAACL HLT 2010 (2010)
16. Pitichotchokphokhin, P., Chuangkrud, P., Kalakan, K., Suntisrivaraporn, B., Leelanupab, T., Kanungsukkasem, N.: Discover underlying topics in thai news articles: A comparative study of probabilistic and matrix factorization approaches. In: ECTI-CON 2020 (2020)

17. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. WSDM '15 (2015)
18. Wang, Z., Ma, L., Zhang, Y.: A hybrid document feature extraction method using latent dirichlet allocation and word2vec. In: DSC 2016 (2016)