A Hidden Topic-based Framework towards Building Applications with Short Web Documents

Xuan-Hieu Phan, Cam-Tu Nguyen, Dieu-Thu Le, Le-Minh Nguyen, Susumu Horiguchi, and Quang-Thuy Ha

***Abstract*— This paper introduces a hidden topic-based framework for processing short and sparse documents (e.g., search result snippets, product descriptions, book/movie summaries, and advertising messages) on the Web. The framework focuses on solving two main challenges posed by these kinds of documents: (1) data sparseness and (2) synonyms/homonyms. The former leads to the lack of shared words and contexts among documents while the latter are big linguistic obstacles in natural language processing (NLP) and information retrieval (IR). The underlying idea of the framework is that common hidden topics discovered from large external datasets (universal datasets), when included, can make short documents less sparse and more topic-oriented. Furthermore, hidden topics from universal datasets help handle unseen data better. The proposed framework can also be applied for different natural languages and data domains. We carefully evaluated the framework by carrying out two experiments for two important online applications (Web search result classification and matching/ranking for contextual advertising) with large-scale universal datasets and we achieved significant results.**

***Index Terms*— Web mining, hidden topic analysis, sparse data, classification, matching, ranking, contextual advertising**

I. Introduction

With the explosion of e-commerce, online publishing, communication, and entertainment, Web data has become available in many different forms, genres, and formats which are much more diverse than ever before. Various kinds of data are generated everyday: queries and questions input by Web search users; Web snippets returned by search engines; Web logs generated by Web servers; chat messages by instant messengers; news feed produced by RSS technology; blog posts and comments by users on a wide spectrum of online forums, e-communities, and social networks; product descriptions and customer reviews on a huge number of e-commercial sites; and online advertising messages from a large number of advertisers.

However, this data diversity has posed new challenges to Web Mining and IR research. Two main challenges we are going to address in this study are (1) short and sparse data problem and (2) synonyms and homonyms. Unlike normal documents, short and sparse documents are usually noisier, less topic-focused, and much shorter, that is, they consist of from a dozen words to a few sentences. Because of the short length, they do not provide enough word co-occurrence patterns or shared contexts for a good similarity measure. Therefore, normal machine learning methods usually fail to achieve the desired accuracy due to the data sparseness. Another problem, which is also likely to happen when we, for instance, train a classification model on sparse data, is that the model has limitations in predicting previously unseen documents due to the fact that the training and the future data share few common features. The latter, i.e., synonyms and homonyms, are natural linguistic phenomena which NLP and IR researchers commonly find difficult to cope with. It is even more difficult with short and sparse data as well as processing models built on top of them. Synonym, that is, two or more different words have similar meanings, causes difficulty in connecting two semantically related documents. For example, the similarity between two (short) documents containing *football* and *soccer* can be zero despite the fact that they can be very relevant. Homonym, on the other hand, means a word can have two or more different meanings. For example, *security* might appear in three different contexts: *national security* (politics), *network security* (information technology), and *security market* (finance). Therefore, it is likely that one can unintentionally put an advertising message about finance on a Web page about politics or technology. These problems, both synonyms and homonyms, can be two of the main sources of error in classification, clustering, and matching, particularly for online contextual advertising ([10], [13], [26], [32], [39], Google AdSense) where we need to put the "right" ad messages on the "right" Web pages in order to attract user attention.

For better retrieving, classifying, clustering, and matching on these kinds of short documents, one can think of a more elegant document representation method beyond vector space model [34]. Query expansion in IR [29] helps overcome the synonym problem in order to improve retrieval precision and recall. It aims at retrieving more relevant and better documents by expanding (i.e., representing) user queries with additional terms using a concept-based thesaurus, word co-occurrence statistics, query logs, and relevance feedback. Latent semantic analysis (LSA) [15], [27] provides a mathematical tool to map vector space model into a more compact space in order to solve synonyms and perform dimensionality reduction. Some studies use clustering as a means to group related words before classification and matching [1], [5], [17]. For matching between short texts, many studies acquire additional information from the Web and search engines [8], [30], [33], [40], [18]. Other studies use taxonomy, ontology, and knowledge base to represent the semantic correlation between words for better classification or clustering.

In this paper, we come up with a general framework for building applications on short Web documents that helps overcome the above challenges by utilizing hidden topics discovered from large-scale external document collections (i.e., universal datasets). The main idea behind the framework is that for each application (e.g., classification, clustering, or contextual advertising), we collect a very large universal dataset, and then build a model on both a small set of annotated data (if available) and a rich set of hidden

• Xuan-Hieu Phan, Cam-Tu Nguyen, and Susumu Horiguchi are with the Graduate School of Information Sciences, Tohoku University, Japan
• Dieu-Thu Le is with Laboratoire Lorrain de Recherche en Informatique et ses Applications, Nancy Universite, France
• Le-Minh Nguyen is with the Graduate School of Information Science, Japan Advanced Institute of Science and Technology, Japan
• Quang-Thuy Ha is with ColTech, Vietnam National University, Hanoi
• This paper is an extension of a shorter version at WWW2008 [31]

Digital Object Indentifier 10.1109/TKDE.2010.27

topics discovered from the universal dataset. These hidden topics, once incorporated into short and sparse documents, will make them less sparse, more topic-focused, and thus giving a better similarity measure between the documents for more accurate classification, clustering, and matching/ranking. Topics inferred from a global data collection like universal dataset help highlight and guide semantic topics hidden in the documents in order to handle synonyms/homonyms, providing a means to build smart Web applications like semantic search, question-answering, and contextual advertising. In general, our main contributions behind this framework are threefold:

- We demonstrate that hidden topic-based approach can be a right solution to sparse data and synonym/homonym problems.
- We show that the framework is a suitable method to build online applications with limited resources. In this framework, universal datasets can be gathered easily because huge document collections are widely available on the Web. By incorporating hidden topics from universal datasets, we can significantly reduce the need of annotated data that is usually expensive and time-consuming to prepare. In this sense, our framework is an alternative to semi-supervised learning [9] because it also effectively takes advantage of external data to improve the performance.
- We empirically show that our framework is highly practical towards building Web applications. We evaluated our framework by carrying out two important experiments/applications: (**a**) Web search domain classification and (**b**) Matching/ranking for online advertising. The first was built upon a universal dataset of more than 30 million words from Wikipedia (English) and the second was with more than 5.5 million words from an online news collection - VnExpress (Vietnamese). The experiments not only show how the framework works with data sparseness, synonym/homonym problems but also demonstrate its flexibility in processing various sorts of Web data, different natural languages, and data domains.

The rest of the paper is organized as follows. Section II reviews some related work. Section III proposes the general framework of classification and contextual matching with hidden topics. Section IV introduces some of the hidden topic analysis models with an emphasis on latent Dirichlet allocation (LDA). Section V describes the topic analysis of large-scale text/Web data collections that serve as universal datasets in the framework. Section VI gives more technical details about how to build a text classifier with hidden topics. Section VII describes how to build a matching and ranking model with hidden topics for online contextual advertising. Section VIII carefully presents two evaluation tasks, the experimental results, and result analysis. Finally, important conclusions are given in Section IX.

II. Related Work

There have been a considerable number of related studies that focused on short and sparse data and attempted to find out a suitable method of representation for the data in order to get a better classification, clustering, and matching performance. In this section, we give a short introduction of several studies that we found most relevant to our work.

The first group of studies focused on the similarity between very short texts. Bollegala et al. 2007 [8] used search engines to get the semantic relatedness between words. Sahami & Heilman 2006 [33] also measured the relatedness between text snippets by using search engines and a similarity kernel function. Metzeler et al. 2007 [30] evaluated a wide range of similarity measures for short queries from Web search logs. Yih & Meek 2007 [40] considered this problem by improving Web-relevance similarity and the method in [33]. Gabrilovich & Markovitch 2007 [18] computed semantic relatedness using Wikipedia concepts.

Prior to recent topic analysis models, word clustering algorithms were introduced to improve text categorization in different ways. Baker & McCallum 1998 [1] attempted to reduce dimensionality by class distribution-based clustering. Bekkerman et al. 2003 [5] combined distributional clustering of words and SVMs. Dhillon & Modha 2001 [17] introduced spherical k-means for clustering sparse text data.

"Text categorization by boosting automatically extracted concepts" by Cai & Hoftmann 2003 [11] is probably the study most related to our framework. Their method attempts to analyze topics from data using probabilistic LSA (pLSA) and uses both the original data and resulting topics to train two different weak classifiers for boosting. The difference is that they extracted topics only from the training and test data while we discover hidden topics from external large-scale data collections. In addition, we aim at processing short and sparse text and Web segments rather than normal text documents. Another related work used topic-based features to improve the word sense disambiguation by Cai et al. 2007 [12].

The success of sponsored search for online advertising has motivated IR researchers to study content match in contextual advertising. Thus, one of the earliest studies in this area was originated from the idea of extracting keywords from Web pages. Those representative keywords will then be matched with advertisements [39]. While extracting keywords from Web pages in order to compute the similarity with ads is still controversial, Andrei Broder et al. [10] proposed a framework for matching ads based on both semantic and syntactic features. For semantic features, they classified both Web pages and ads into the same large taxonomy with 6000 nodes. Each node contains a set of queries. For syntactic features, they used the TF-IDF score and section score (title, body or bid phrase section) for each term of Web pages or ads. Our framework also tries to discover the semantic relations of Web pages and ads, but instead of using a classifier with a large taxonomy, we use hidden topics discovered automatically from an external dataset. It does not require any language-specific resources, but simply takes advantage of a large collection of data, which can be easily gathered on the Internet.

One challenge of contextual matching task is the difference between the vocabularies of Web pages and ads. Ribeiro-Neto et al. [32] focused on solving this problem by using additional pages. It is similar to ours in the idea of expanding Web pages with external terms to decrease the distinction between their vocabularies. However, they determined added terms from other similar pages by means of a Bayesian model. Those extended terms can appear in ad's keywords and potentially improve the overall performance of the framework. Their experiments have proved that when decreasing the vocabulary distinction between Web pages and ads, we can find better ads for a target page.

Following the former study [32], Lacerda et al. [26] tried to improve the ranking function based on Genetic Programming. Given the importance of different features, such as term and document frequencies, document length and collection's size, they used machine learning to produce a matching function to optimize the relevance between the target page and ads. It was represented

as a tree composed of operators and logarithm as nodes and features as leaves. They used a set of data for training and a set for evaluating from the same data set used in [32] and recorded a better gain over the best method described in [32] of 61.7%.

III. The General Framework

In this section, we give a general description of the proposed framework: *classifying, clustering, and matching with hidden topics discovered from external large-scale data collections*. It is general enough to be applied to different tasks, and among them we take two problems: *document classification* and *online contextual advertising* as the demonstration.

Document classification, also known as text categorization, has been studied intensively during the past decade. Many learning methods, such as k nearest neighbors (k-NN), Naive Bayes, maximum entropy, and support vector machines (SVMs), have been applied to a lot of classification problems with different benchmark collections (Reuters-21578, 20Newsgroups, WebKB, etc.) and achieved satisfactory results [2], [36]. However, our framework mainly focuses on text representation and how to enrich short and sparse texts to enhance classification accuracy.

Online contextual advertising, also known as contextual match or targeted advertising, has emerged recently and become an essential part of online advertising. Since its birth more than a decade ago, online advertising has grown quickly and become more diverse in both its appearance and the way it attracts Web users' attention. According to the Interactive Advertising Bureau (IAB) [21], Internet advertising revenues reached $5.8 billion for the first quarter of 2008, 18.2% increase over the same period in 2007. Its growth is expected to continue as consumers spend more and more time online. One important observation is that the relevance between target Web pages and advertising messages is a significant factor to attract online users and customers [13], [37]. In contextual advertising, ad messages are delivered based on the content of the Web pages that users are surfing. It can therefore provide Internet users with information they are interested in and allow advertisers to reach their target customers in a non-intrusive way. In order to suggest the "right" ad messages, we need efficient and elegant contextual ad matching and ranking techniques.

Different from sponsored search, in which advertising are chosen depending on only the keywords provided by users, contextual ad placement depends on the whole content of a Web page. Keywords given by users are often condensed and reveal directly the content of the users' concerns, which make it easier to understand. Analyzing Web pages to capture the relevance is a more complicated task. Firstly, as words can have multiple meanings and some words in the target page are not important, they can lead to mismatch in lexicon-based matching method. Moreover, a target page and an ad can still be a good match even when they share no common words or terms. Our framework, that can produce high quality match that takes advantage of hidden topics analyzed from large-scale external datasets, should be a suitable solution to the problem.

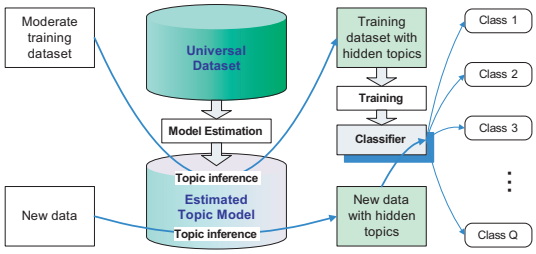
A. Classification with Hidden Topics

Given a small training data set $\mathbf{D} = \{(d_1, c_1), (d_2, c_2), \ldots, (d_n, c_n)\}$ consisting of n short and sparse documents d_i and their class labels c_i $(i = 1..n)$; and $\mathbf{W} = \{w_1, w_2, \ldots, w_m\}$ be a large scale data collection containing m unlabeled documents w_i $(i = 1..m)$. Note that the documents in $\mathbf{W}$ are usually longer and not required to have the same format with the documents in $\mathbf{D}$. Our approach provides a framework to gain additional knowledge from $\mathbf{W}$ in terms of hidden topics to modify and enrich the training set $\mathbf{D}$ in order to build a better classification model. Here, we call $\mathbf{W}$ "**universal dataset**" since it is large and diverse enough to cover a lot of information (e.g., words/topics) regarding the classification task. The whole framework of "learning to classify with hidden topics" is depicted in Figure 1. The framework consists of five sub-tasks: (a) collecting universal dataset $\mathbf{W}$, (b) carrying out topic analysis for $\mathbf{W}$, (c) preparing labeled training data, (d) performing topic inference for training & test data, and (e) building the classifier.

Among the five steps, choosing a right universal dataset (a) is probably the most important. First, the universal dataset, as its name implies, must be large and rich enough to cover a lot of words, concepts, and topics which are relevant to the classification problem. Second, this dataset should be consistent with the training and future unseen data that the classifier will work with. This means that the nature of universal data (e.g., patterns, statistics, and co-occurrence of them) should be observed by humans to determine whether or not the potential topics analyzed from this data can help to make the classifier more discriminative. This will be discussed more in Section V where we analyze two large-scale text & Web collections for solving two classification problems. Step (b), doing topic analysis for the universal dataset, is performed by using one of the well-known hidden topic analysis models such as pLSA or LDA. We chose LDA because this model has a more complete document generation assumption. LDA will be briefly introduced in Section IV. The analysis process of Wikipedia is described in detail in Section V.

In general, building a large amount of labeled training data for text classification is a labor-intensive and time-consuming task. Our framework can avoid this by requiring a moderate size or even small size of labeled data (c). One thing needing more attention is that words/terms in this dataset should be relevant to as many hidden topics as possible. This is to ensure that most hidden topics are incorporated into the classifier. Therefore, in spite of small size, the labeled training data should be balanced among topics. The experiments in Section VIII will show how well the framework can work with small size of training data.

Fig. 1. Framework of learning to classify sparse text/Web with hidden topics



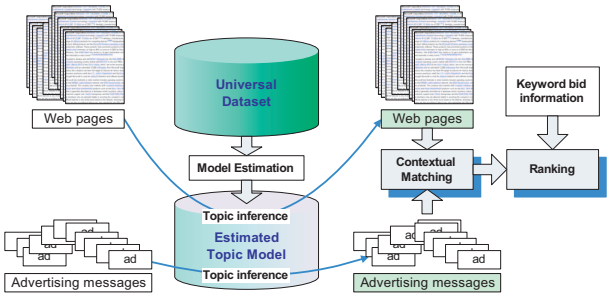
(a) *Choosing an appropriate "universal dataset"*
(b) *Doing topic analysis for the universal dataset*
(c) *Building a moderate size labeled training dataset*
(d) *Doing topic inference for training and future data*
(e) *Building the classifier*

Topic inference for training and future unseen data (d) is another important issue. This depends on not only LDA but also which machine learning technique we choose to train the

classifier. This will be discussed more detailed in Section VI-B. Building a classifier (e) is the final procedure. After doing topic inference for training data, this step is similar to any other training process to build a text classifier. In this work, we used maximum entropy (MaxEnt) for building classifiers. Section VI will give a more detailed discussion about this.

B. Contextual Advertising: Matching/Ranking with Hidden Topics

Fig. 2. Framework of page-ad matching & ranking with hidden topics



(a) *Choosing an appropriate "universal dataset"*
(b) *Doing topic analysis for the universal dataset*
(c) *Doing topic inference for Web pages and ads*
(d) *Page-Ad Matching and Ranking*

In this section, we present our general framework for contextual page-ad matching and ranking with hidden topics discovered from external large-scale data collections.

Given a set of n target Web pages $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$, and a set of m ad messages (ads) $\mathbf{A} = \{a_1, a_2, \ldots, a_m\}$. For each Web page p_i, we need to find a corresponding ranking list of ads: $\mathbf{A}_i = \{a_{i1}, a_{i2}, \ldots, a_{im}\}, i \in 1..n$ such that more relevant ads will be placed higher in the list. These ads are ranked based on their relevance to the target page and the keyword bid information. However, in the scope of our work, we only take linguistic relevance into consideration and assume that all ads have the same priority, i.e., the same bid amount.

As depicted in Figure 2, the first important thing to consider in this framework is collecting an external large-scale document collection (a) which is called universal dataset. To take the best advantage of it, we need to find an appropriate universal dataset for the Web pages and ad messages. First, it must be large enough to cover words, topics, and concepts in the domains of Web pages and ads. Second, its vocabularies must be consistent with those of Web pages and ads, so that it will make sure topics analyzed from this data can overcome the vocabulary impedance of Web pages and ads. The universal dataset should also be preprocessed to remove noise and stop words before analysis to get better results. The result of step (b), hidden topic analysis, is an estimated topic model that includes hidden topics discovered from the universal dataset and the distributions of topics over terms. Steps (a) and (b) will be presented more details in section V and subsection V-B. After step (b), we can again do topic inference for both Web pages and ads based on this model to discover their meanings and topic focus (c). This information will be integrated into the corresponding Web pages or ads for matching and ranking (d). Both steps (c) and (d) will be discussed more in section VII.

IV. Hidden Topic Analysis Models

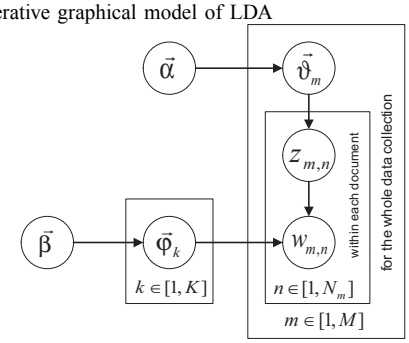
Latent Dirichlet Allocation (LDA), first introduced by Blei et al. [6], is a probabilistic generative model that can be used to estimate the multinomial observations by unsupervised learning. With respect to topic modeling, LDA is a method to perform so-called latent semantic analysis (LSA). The intuition behind LSA is to find the latent structure of "topics" or "concepts" in a text corpus. The term LSA has been coined by Deerwester et al. [15] who empirically showed that the co–occurrence (both direct and indirect) of terms in text documents can be used to recover this latent topic structure. In turn, latent-topic representation of text allows to model linguistic phenomena like synonymy and polysemy. This allows IR systems to represent text in a way suitable for matching user queries on a semantic level rather than by lexical occurrence. LDA is closely related to the probabilistic latent semantic analysis (pLSA) by Hofmann [24], a probabilistic formulation of LSA. However, it has been pointed out that LDA is more complete than pLSA in such a way that it follows a full generation process for document collection [6], [20], [23]. Models like pLSA, LDA, and their variants have had more successful applications in document & topic modeling [6], [20], dimensionality reduction for text categorization [6], collaborative filtering [25], ad hoc IR [38], and digital library [7].

A. Latent Dirichlet Allocation (LDA)

LDA is a generative graphical model as shown in Figure IV-A. It can be used to model and discover underlying topic structures of any kind of discrete data in which text is a typical example. LDA was developed based on an assumption of document generation process depicted in both Figure IV-A and Table I. This process can be interpreted as follows.

In LDA, a document $\overrightarrow{w}_m = \{w_{m,n}\}_{n=1}^{N_m}$ is generated by first picking a distribution over topics $\overrightarrow{\vartheta}_m$ from a Dirichlet distribution ($Dir(\overrightarrow{\alpha})$), which determines topic assignment for words in that document. Then the topic assignment for each word placeholder $[m,n]$ is performed by sampling a particular topic $z_{m,n}$ from multinomial distribution $Mult(\overrightarrow{\vartheta}_m)$. Finally, a particular word $w_{m,n}$ is generated for the word placeholder $[m,n]$ by sampling from multinomial distribution $Mult(\overrightarrow{\varphi}_{z_{m,n}})$.

Fig. 3. Generative graphical model of LDA



From the generative graphical model depicted in Figure IV-A, we can write the joint distribution of all known and hidden variables given the Dirichlet parameters as follows.

$$p(\overrightarrow{w}_m, \overrightarrow{z}_m, \overrightarrow{\vartheta}_m, \Phi | \overrightarrow{\alpha}, \overrightarrow{\beta}) = p(\Phi|\overrightarrow{\beta}) \prod_{n=1}^{N_m} p(w_{m,n}|\overrightarrow{\varphi}_{z_{m,n}}) p(z_{m,n}|\overrightarrow{\vartheta}_m) p(\overrightarrow{\vartheta}_m|\overrightarrow{\alpha})$$

And the likelihood of a document $\overrightarrow{w}_m$ is obtained by integrating over $\overrightarrow{\vartheta}_m$, Φ and summing over $\overrightarrow{z}_m$ as follows.

TABLE I
GENERATION PROCESS FOR LDA

for all topics $k \in [1, K]$ **do**
  sample mixture components $\overrightarrow{\varphi}_k \sim Dir(\overrightarrow{\beta})$
end for
for all documents $m \in [1, M]$ **do**
  sample mixture proportion $\overrightarrow{\vartheta}_m \sim Dir(\overrightarrow{\alpha})$
  sample document length $N_m \sim Poiss(\xi)$
  for all words $n \in [1, N_m]$ **do**
    sample topic index $z_{m,n} \sim Mult(\overrightarrow{\vartheta}_m)$
    sample term for word $w_{m,n} \sim Mult(\overrightarrow{\varphi}_{z_{m,n}})$
  end for
end for

- M: the total number of documents
- K: the number of (hidden/latent) topics
- V: vocabulary size
- $\overrightarrow{\alpha}$, $\overrightarrow{\beta}$: Dirichlet parameters
- $\overrightarrow{\vartheta}_m$: topic distribution for document m
- $\Theta = \{\overrightarrow{\vartheta}_m\}_{m=1}^{M}$: a $M \times K$ matrix
- $\overrightarrow{\varphi}_k$: word distribution for topic k
- $\Phi = \{\overrightarrow{\varphi}_k\}_{k=1}^{K}$: a $K \times V$ matrix
- N_m: the length of document m
- $z_{m,n}$: topic index of nth word in document m
- $w_{m,n}$: a particular word for word placeholder [m, n]

$$p(\overrightarrow{w}_m|\overrightarrow{\alpha}, \overrightarrow{\beta}) = \int\int p(\overrightarrow{\vartheta}_m|\overrightarrow{\alpha})p(\Phi|\overrightarrow{\beta}) \cdot \prod_{n=1}^{N_m} p(w_{m,n}|\overrightarrow{\vartheta}_m, \Phi)d\Phi d\overrightarrow{\vartheta}_m$$

Finally, the likelihood of the whole data collection $\mathcal{W} = \{\overrightarrow{w}_m\}_{m=1}^{M}$ is product of the likelihoods of all documents:

$$p(\mathcal{W}|\overrightarrow{\alpha}, \overrightarrow{\beta}) = \prod_{m=1}^{M} p(\overrightarrow{w}_m|\overrightarrow{\alpha}, \overrightarrow{\beta}) \quad (1)$$

B. LDA Estimation with Gibbs Sampling

Estimating parameters for LDA by directly and exactly maximizing the likelihood of the whole data collection in (1) is intractable. The solution to this is to use approximate estimation methods like Variational Methods [6] and Gibbs Sampling [20]. Gibbs Sampling is a special case of Markov-chain Monte Carlo (MCMC) [19] and often yields relatively simple algorithms for approximate inference in high-dimensional models like LDA [23].

The first use of Gibbs Sampling for estimating LDA is reported in [20] and a more comprehensive description of this method is from the technical report [23]. One can refer to these papers for a better understanding of this sampling technique. Here, we only show the most important formula that is used for topic sampling for words. Let $\overrightarrow{w}$ and $\overrightarrow{z}$ be the vectors of all words and their topic assignment of the whole data collection W. The topic assignment for a particular word depends on the current topic assignment of all the other word positions. More specifically, the topic assignment of a particular word t is sampled from the following multinomial distribution.

$$p(z_i = k|\overrightarrow{z}_{\neg i}, \overrightarrow{w}) = \frac{n_{k,\neg i}^{(t)} + \beta_t}{[\sum_{v=1}^{V} n_k^{(v)} + \beta_v] - 1} \frac{n_{m,\neg i}^{(k)} + \alpha_k}{[\sum_{j=1}^{K} n_m^{(j)} + \alpha_j] - 1} \quad (2)$$

where $n_{k,\neg i}^{(t)}$ is the number of times the word t is assigned to topic k except the current assignment; $\sum_{v=1}^{V} n_k^{(v)} - 1$ is the total number of words assigned to topic k except the current assignment; $n_{m,\neg i}^{(k)}$ is the number of words in document m assigned to topic k except the current assignment; and $\sum_{j=1}^{K} n_m^{(j)} - 1$ is the total number of words in document m except the current word t. In normal cases, Dirichlet parameters $\overrightarrow{\alpha}$, and $\overrightarrow{\beta}$ are symmetric, that is, all α_k (k = 1..K) are the same, and similar for β_v (v = 1..V).

After finishing Gibbs Sampling, two matrices Φ and Θ are computed as follows.

$$\varphi_{k,t} = \frac{n_k^{(t)} + \beta_t}{\sum_{v=1}^{V} n_k^{(v)} + \beta_v} \quad (3)$$

$$\vartheta_{m,k} = \frac{n_m^{(k)} + \alpha_k}{\sum_{j=1}^{K} n_m^{(j)} + \alpha_j} \quad (4)$$

V. LARGE-SCALE TEXT AND WEB COLLECTIONS AS UNIVERSAL DATASETS

A. Hidden Topic Analysis of Wikipedia Data

Today, Wikipedia has been known as the richest online encyclopedia written collaboratively by a large number of contributors around the world. A huge number of documents available in various languages and placed in a nice structure (with consistent formats and category labels) do inspire the WWW, IR, and NLP research communities to think of using it as a huge corpus [16]. Some previous researches have utilized it for short text clustering [3], measuring relatedness [18], and topic identification [35].

1) Data Preparation: Since Wikipedia covers a lot of concepts and domains, it is reasonable to use it as a universal dataset in our framework for classifying and clustering short and sparse text/Web. To collect the data, we prepared various *seed* crawling keywords coming from different domains as shown in the following table. For each seed keyword, we ran JWikiDocs[1] to download the corresponding Wikipedia page and crawl relevant pages by following outgoing hyperlinks. Each crawling transaction is limited by the total number of download pages or the maximum depth of hyperlinks (usually 4).

After crawling, we got a total of 3.5GB with more than 470,000 Wikipedia documents. Because the outputs of different crawling transactions share a lot of common pages, we removed these duplicates and obtained more than 70,000 documents. After removing HTML tags, noisy text and links, rare (threshold = 30) and stop words, we obtained the final dataset as in Table II.

TABLE II
WIKIPEDIA AS THE UNIVERSAL DATASET

Topic-oriented keywords for crawlingWikipedia
Arts architecture, fine art, dancing, fashion, film, music ...
Business advertising, e-commerce, finance, investment ...
Computers hardware, software, database, multimedia ...
Education course, graduate, professor, university ...
Engineering automobile, telecommunication, civil eng. ...
Entertainment book, music, movie, painting, photos ...
Health diet, therapy, healthcare, treatment, nutrition ...
Mass-media news, newspaper, journal, television ...
Politics government, legislation, party, regime, military ...
Science biology, physics, chemistry, ecology, laboratory ...
Sports baseball, cricket, football, tennis, olympic games ...
Misc. association, development, environment ...
Statistics of the crawled Wikipedia data
Raw data: 3.5GB; |docs| = 471,177
Preprocessing: removing duplicate docs, HTML tags, navigation links, stop and rare words
Final data: 240MB; |docs| = 71,986; |paragraphs| = 882,376; |vocabulary| = 60,649; |total words| = 30,492,305

[1] JWikiDocs: http://jwebpro.sourceforge.net

Fig. 4. Most likely words of some sample topics of Wikipedia data. See the complete results online at: http://gibbslda.sourceforge.net/wikipedia-topics.txt

T0: medical health medicine care practice patient training treatment patients hospital surgery clinical physicians physician hospitals doctors therapy physical nursing doctor ...
T1: memory intel processor instruction processors cpu performance instructions architecture hardware data address core cache computer processing operating program ...
T4: signal radio frequency signals digital transmission channel antenna frequencies receiver communication transmitter analog modulation transmitted mhz data channels ...
T10: theory quantum universe physics mechanics particles matter particle relativity einstein model space physical light classical field theories principle energy fundamental ...
T18: economic trade economy world government growth countries country industry foreign production sector gdp development domestic billion industrial market policy nations ...
T19: film films movie production movies director cinema studio hollywood released pictures picture studios directed motion release shot sound scene actors ...
T20: party election vote elections parties voting votes candidate candidates majority political voters seats electoral democratic elected opposition coalition government ballot ...
T22: tax income taxes pay paid rate revenue taxation government benefit plan sales benefits rates value plans money cost property federal ...
T27: philosophy philosophers world philosophical knowledge mind reality aristotle existence nature plato ideas experience philosopher view consciousness kant physical idea ...
T28: space function functions vector linear theory geometry matrix equations mathematics equation field theorem algebra mathematical spaces differential product continuous ...
T33: insurance debt risk rate credit bonds pay loss loan cash policy payment bond money paid rates loans cost payments financial ...
T34: university college degree students universities school research academic student degrees campus colleges education graduate professor master institute institutions ...
T38: law act rights laws court constitution federal united legal government supreme legislation amendment civil constitutional congress public process justice power ...
T45: network networks protocol server data internet client ip nodes node connection servers protocols address packet layer connections service routing link ...
T55: government house parliament minister prime president power executive elected office council constitution assembly appointed powers head cabinet parliamentary ...
T57: cell cells protein proteins membrane molecules amino enzymes enzyme structure binding acids process bacteria acid cellular receptor antibodies receptors atp ...
T60: radio television tv stations broadcast channel news network station cable broadcasting bbc satellite programming channels service media networks broadcasts program ...
T62: music jazz dance folk blues songs musicians style musical styles traditional american song rhythm country pop performers artists played dances ...
T64: gold currency dollar coins silver value money coin issued exchange euro inflation monetary rate pound currencies paper standard dollars mint ...
T73: internet online users site com content sites community web website user virtual information websites people software media personal forums yahoo ...
T81: art artists painting paintings artist style arts movement artistic sculpture museum painted aesthetic abstract visual painters figures architecture beauty gallery ...
T84: race sports sport racing olympic events world event competition races games team golf course olympics track international championship teams formula ...
T93: military army service officers forces force officer rank training command war armed united personnel units air soldiers ranks corps navy ...
T98: bc ancient egyptian egypt civilization period culture bronze bce age king city maya archaeological stone cities egyptians temple millennium discovered ...
T101: magic harry potter magical house witch book witchcraft wizard witches magician books people spell wizards hogwarts rowling black paranormal phoenix ...
T103: card cards stores store chain department items retail customer customers shopping credit chains service retailers cash item shop merchant target ...
T104: software windows file microsoft operating version user files os applications linux source system mac versions application users released code release ...
T107: market price stock value exchange trading markets prices sell options buy spread index stocks risk selling trade features shares contracts ...
T137: bank money banks account credit financial banking central accounts reserve balance funds federal savings services deposit loans transactions deposits commercial ...
T152: economics economic value market theory price demand production capital economy cost economists costs prices marginal utility money output labor inflation ...
T199: distribution probability test random sample variables statistical variable data error analysis function value mean tests inverse statistics values hypothesis correlation ...

2) Analysis and Outputs: We estimated many LDA models for the Wikipedia data using GibbsLDA++[2], our C/C++ implementation of LDA using Gibbs Sampling. The number of topics ranges from 10, 20 ... to 100, 150, and 200. The hyperparameters alpha and beta were set to 0.5 and 0.1, respectively. Some sample topics from the model of 200 topics are shown in Figure 4. We observed that the analysis outputs (topic-document and topic-word distributions) satisfy our expectation. These LDA models will be used for topic inference to build Web search domain classifiers in Section VIII.

B. Hidden Topic Analysis of Online News Collection

TABLE III
VNEXRESS NEWS COLLECTION SERVING AS "UNIVERSAL DATASET" FOR CONTEXTUAL ADVERTISING

After removing html, doing sentence and word segmentation: $size \approx 219M, \|docs\| = 40,328$
After filtering and removing non-topic oriented words: $size \approx 53M, \|docs\| = 40,268$ $\|words\| = 5,512,251;\ \|vocabulary\| = 128,768$

This section brings an in-detail description of hidden topic analysis of a large-scale Vietnamese news collection that serves as a "universal dataset" in the general framework for contextual advertising mentioned earlier in Section III-B. With the purpose of using a large scale dataset for Vietnamese contextual advertising, we choose VnExpress[3] as the universal dataset for topic analysis. VnExpress is one of the highest ranking e-newspaper corporations in Vietnam, thus containing a large number of articles in many topics in daily life. For this reason, it is a suitable data collection for advertising areas.

This news collection includes different topics, such as Society, International news, Lifestyle, Culture, Sports, Science, etc. We crawled 220 Megabyte of approximately 40,000 pages using Nutch[4]. We then performed some preprocessing steps (HTML removal, sentence/word segmentation, stop words and noise removal, etc.) and finally got more than 50 Megabyte plain text. See Table III for the details of this data collection.

We performed topic analysis for this news collection using GibbsLDA++ with different number of topics (60, 120, and 200). Figure 5 shows several sample hidden topics discovered from VnExpress. Each column (i.e., each topic) includes Vietnamese words in that topic and their corresponding translations in English in the parentheses. These analysis outputs will be used to enrich both target Web pages and advertising messages (ads) for matching and ranking in contextual advertising. This will be discussed more detailed in Section VII.

VI. BUILDING CLASSIFIERS WITH HIDDEN TOPICS

Building a classifier after topic analysis for the universal dataset includes three main steps. First, we choose one from different learning methods, such as Naive Bayes, maximum entropy (MaxEnt), SVMs, etc. Second, we integrate hidden topics into the training, test, or future data according to the data representation of the chosen learning technique. Finally, we train the classifier on the integrated training data.

A. Choosing Machine Learning Method

Many traditional classification methods, such as k–NN, Decision Tree, Naive Bayes, and more recent advanced models like MaxEnt, SVMs can be used in our framework. Among them, we chose MaxEnt [4] because of two main reasons. First, MaxEnt is robust and has been applied successfully to a wide range of NLP tasks, such as part-of-speech (POS) tagging, named entity recognition (NER), parsing, etc. It even performs better than SVMs [22] and others in some particular cases, such as classifying sparse data. Second, it is very fast in both training and inference. SVMs is also a good choice because it is powerful. However, the

[2]GibbsLDA++: http://gibbslda.sourceforge.net
[3]VnExpress: The Online Vietnamese News - http://vnexpress.net
[4]Nutch: an open-source search engine, http://lucene.apache.org/nutch

Fig. 5. Sample topics analyzed from VnExpress News Collection. See the complete results online at http://gibbslda.sourceforge.net/vnexpress-200topics.txt

Topic 3	Topic 15	Topic 44	Topic 48	Topic 56	Topic 172
bác_sĩ (doctor)	**thời_trang** (fashion)	**thiết_bị** (equipment)	**chứng_khoán** (stock)	**bánh** (cake)	**thẻ** (card)
bệnh_viện (hospital)	**người_mẫu** (model)	**sản_phẩm** (product)	**công_ty** (company)	**mcdonald** (McDonald)	**khoá** (lock)
thuốc (medicine)	**mặc** (wear)	**máy** (machine)	**đầu_tư** (investment)	**thịt** (meat)	**rút** (withdraw)
bệnh (disease)	**trang_phục** (clothes)	**màn_hình** (screen)	**ngân_hàng** (bank)	**pizza** (pizza)	**chủ** (owner)
phẫu_thuật (surgery)	**thiết_kế** (design)	**công_nghệ** (technology)	**cổ_phần** (joint-stock)	**ba_tê** (pate)	**chìa** (key)
điều_trị (treatment)	**đẹp** (beautiful)	**điện_thoại** (telephone)	**thị_trường** (market)	**bánh_mì** (bread)	**thẻ_tín_dụng** (credit card)
bệnh_nhân (patient)	**váy** (dress)	**hãng** (company)	**giao_dịch** (transaction)	**bánh_ngọt** (pie)	**atm** (ATM)
y_tế (medical)	**sưu_tập** (collection)	**sử_dụng** (use)	**đồng** (VND)	**cửa_hàng** (shop)	**tín_dụng** (credit)
ung_thư (cancer)	**mang** (wear)	**thị_trường** (market)	**mua** (buy)	**xúc_xích** (hot dog)	**thanh_toán** (pay)
tình_trạng (condition)	**phong_cách** (style)	**usd** (USD)	**phát_hành** (publish)	**kem** (ice-cream)	**visa** (visa)
cơ_thể (body)	**quần_áo** (costume)	**pin** (battery)	**niêm_yết** (post)	**khai_trương** (open)	**tối_thiểu** (minimum)
sức_khoẻ (health)	**nổi_tiếng** (famous)	**cho_phép** (allow)	**bán** (sell)	**nguội** (cold)	**mastercard**
đau (hurt)	**quần** (trousers)	**samsung** (Samsung)	**tài_chính** (finance)	**hamburger** (hamburger)	**phát_hành** (release)
gây (cause)	**trình_diễn** (perform)	**di_động** (mobile)	**đấu_giá** (auction)	**thịt** (meat)	**trả_nợ** (pay debt)
khám (examine)	**thích** (like)	**sony** (Sony)	**trung_tâm** (center)	**nhà_hàng** (restaurant)	**sẵn_sàng** (ready)
kết_quả (result)	**quyến_rũ** (charming)	**nhạc** (music)	**thông_tin** (information)	**đồ_ăn** (food)	**mật_mã** (password)
căn_bệnh (illness)	**sang_trọng** (luxurious)	**máy_tính** (computer)	**doanh_nghiệp** (business)	**sandwich** (sandwich)	**thường_niên** (annual)
nặng (serious)	**vẻ_đẹp** (beauty)	**hỗ_trợ** (support)	**cổ_đông** (shareholder)	**khẩu_vị** (taste)	**cảnh_giác** (alert)
cho_biết (inform)	**gái** (girl)	**điện_tử** (electronic)	**nhà_đầu_tư** (investor)	**tiệm_bánh** (bakery)	**chủ_thẻ** (card owner)
máu (blood)	**gương_mặt** (figure)	**tính_năng** (feature)	**nhà_nước** (government)	**bảo_đảm** (ensure)	**theo_dõi** (follow)
xét_nghiệm (test)	**siêu** (super)	**kết_nối** (connect)	**tổ_chức** (organization)	**nướng** (grill)	**nhà_băng** (bank)
chữa (cure)	**áo_dài** (aodai)	**thiết_kế** (design)	**triệu** (million)	**bí_quyết** (secret)	**tội_phạm** (criminal)
chứng (trouble)	**giày** (shoes)	**chức_năng** (function)	**quỹ** (budget)	**ngon** (delicious)	**trộm** (steal)

learning and inference speed of SVMs is still a challenge to apply to almost real-time applications.

B. Topic Inference and Integration into Data

Given a set of new documents $\underline{W} = \{\overrightarrow{w}_{\underline{m}}\}_{\underline{m}=1}^{M}$, keep in mind that $\underline{W}$ is different from the *universal dataset* W. For example, W is a collection of Wikipedia documents while $\underline{W}$ is a set of Web search snippets that we need to classify. $\underline{W}$ can be the training, test, or future data. Topic inference for documents in $\underline{W}$ also needs to perform Gibbs Sampling. However, the number of sampling iterations for inference is much smaller than that for the parameter estimation. We observed that about 20 or 30 iterations are enough.

Let $\overrightarrow{w}$ and $\overrightarrow{z}$ be the vectors of all words and their topic assignment in the whole universal dataset W, and $\underline{\overrightarrow{w}}$ and $\underline{\overrightarrow{z}}$ denote the vectors of all words and their topic assignment in the whole new dataset $\underline{W}$. The topic assignment for a particular word t in $\underline{\overrightarrow{w}}$ depends on the current topic assignment for all the other words in $\underline{\overrightarrow{w}}$ and the topic assignment of all words in $\overrightarrow{w}$ as follows.

$$p(\underline{z}_i = k | \underline{\overrightarrow{z}}_{\neg i}, \underline{\overrightarrow{w}}, \overrightarrow{z}, \overrightarrow{w}) = \frac{n_k^{(t)} + \underline{n}_{k,\neg i}^{(t)} + \beta_t}{[\sum_{v=1}^{V} n_k^{(v)} + \underline{n}_k^{(v)} + \beta_v] - 1} \frac{n_{\underline{m},\neg i}^{(k)} + \alpha_k}{[\sum_{j=1}^{K} n_{\underline{m}}^{(j)} + \alpha_j] - 1} \quad (5)$$

where $\underline{n}_{k,\neg i}^{(t)}$ is the number of times the current word t is assigned to topic k within $\underline{\overrightarrow{W}}$ except the current assignment; $\sum_{v=1}^{V} \underline{n}_k^{(v)} - 1$ is the number of words in $\underline{\overrightarrow{W}}$ that are assigned to topic k except the current assignment; $n_{\underline{m},\neg i}^{(k)}$ is the number of words in document $\underline{m}$ assigned to topic k except the current assignment; and $\sum_{j=1}^{K} n_{\underline{m}}^{(j)} - 1$ is the total of words in document $\underline{m}$ except the current word t.

After performing topic sampling, the topic distribution of a new document $\underline{\overrightarrow{w}}_{\underline{m}}$ is $\underline{\overrightarrow{\vartheta}}_{\underline{m}} = \{\vartheta_{\underline{m},1}, \ldots, \vartheta_{\underline{m},k}, \ldots, \vartheta_{\underline{m},K}\}$ where each distribution component is computed as follows.

$$\vartheta_{\underline{m},k} = \frac{n_{\underline{m}}^{(k)} + \alpha_k}{\sum_{j=1}^{K} n_{\underline{m}}^{(j)} + \alpha_j} \quad (6)$$

After doing topic inference, we will integrate the topic distribution $\underline{\overrightarrow{\vartheta}}_{\underline{m}} = \{\vartheta_{\underline{m},1}, \ldots, \vartheta_{\underline{m},k}, \ldots, \vartheta_{\underline{m},K}\}$ and the original document $\underline{\overrightarrow{w}}_{\underline{m}} = \{w_{\underline{m},1}, w_{\underline{m},2}, \ldots, w_{\underline{m},N_{\underline{m}}}\}$ in order that the resulting vector is suitable for the chosen learning technique. This combination is non-trivial because the first vector is a probabilistic distribution while the second is a bag-of-word vector and their importance weights are different. This integration directly influences the learning and classification performance.

Here we describe how we integrate $\underline{\overrightarrow{\vartheta}}_{\underline{m}}$ into $\underline{\overrightarrow{w}}_{\underline{m}}$ to be suitable for building the classifier using MaxEnt. Because MaxEnt requires discrete feature attributes, it is necessary to discretize the probability values in $\underline{\overrightarrow{\vartheta}}_{\underline{m}}$ to obtain topic names. The name of a topic appears once or several times depending on the probability of that topic. For example, a topic with probability in interval [0.05, 0.10) will appear 4 times (denote [0.05, 0.10):4). Here is an example of integrating the topic distribution into its bag-of-word vector to obtain the **snippet1** as shown in Figure 6.

- $\underline{\overrightarrow{w}}_{\underline{m}}$ = {online poker tilt poker money ... card}
- $\underline{\overrightarrow{\vartheta}}_{\underline{m}}$ = {..., $\vartheta_{\underline{m},70} = 0.0208$, ..., $\vartheta_{\underline{m},103} = 0.1125$, ..., $\vartheta_{\underline{m},137} = 0.0375$, ..., $\vartheta_{\underline{m},188} = 0.0125$, ...}
- Applying discretization intervals
- $\underline{\overrightarrow{w}}_{\underline{m}} \cup \underline{\overrightarrow{\vartheta}}_{\underline{m}}$ = **snippet1**, shown in Figure 6

The top part in Figure 6 shows an example of 9 Web search snippets after doing topic inference and integration. Those snippets will be used with a MaxEnt classifier. For other learning techniques like SVMs, we need another integration because SVMs work with numerical vectors.

Inferred hidden topics really make the data more related. This is demonstrated by the middle and the bottom parts in Figure 6. The middle part shows the sparseness among 9 Web snippets in which only a small fraction of words are shared by two or three different snippets. Even some common words, such as "search", "online", and "compare", are not useful (noisy) because they are not related to *business* domain of the 9 snippets. The bottom part visualizes the topics shared among snippets after doing inference and integration. Most shared topics, such as "T22", "T33", "T64", "T73", "T103", "T107", "T152", and specially "T137" make the snippets more related in a semantic way. Refer to Figure 4 to see what these topics are about.

C. Training the Classifier

We train the MaxEnt classifier on the integrated data by using limited memory optimization (L-BFGS) [28]. As shown in recent studies, training using L-BFGS gives high performance in terms of speed and classification accuracy. All MaxEnt classifiers in

Fig. 6. Top: sample Google search snippets (including Wikipedia topics after inference); Middle: visualization of snippet-word co–occurrences; Bottom: visualization of shared topics among snippets after inference

(snippet1) online poker tilt poker money payment processing deposit money tilt poker account visa mastercard credit card atm check debit card *topic:70* *topic:103* *topic:103* *topic:103* *topic:103* *topic:137* *topic:137* *topic:188*

(snippet2) money payment proof broker payment online payment e-gold ebullion liberty reserve web money edinar wire transfer *topic:33* *topic:33* *topic:68* *topic:69* *topic:73* *topic:103* *topic:133* *topic:137* *topic:151*

(snippet3) savings accounts isas investments compare savings isa accounts cash isas access savings investment bonds moneysupermarket com *topic:1* *topic:22* *topic:33* *topic:45* *topic:64* *topic:73* *topic:117* *topic:137* *topic:137* *topic:138* *topic:152* *topic:153* *topic:179*

(snippet4) savings accounts online banking rate apy compare online banking rates savings account features rates apy help online savings *topic:22* *topic:32* *topic:64* *topic:73* *topic:89* *topic:107* *topic:137* *topic:137* *topic:137* *topic:199*

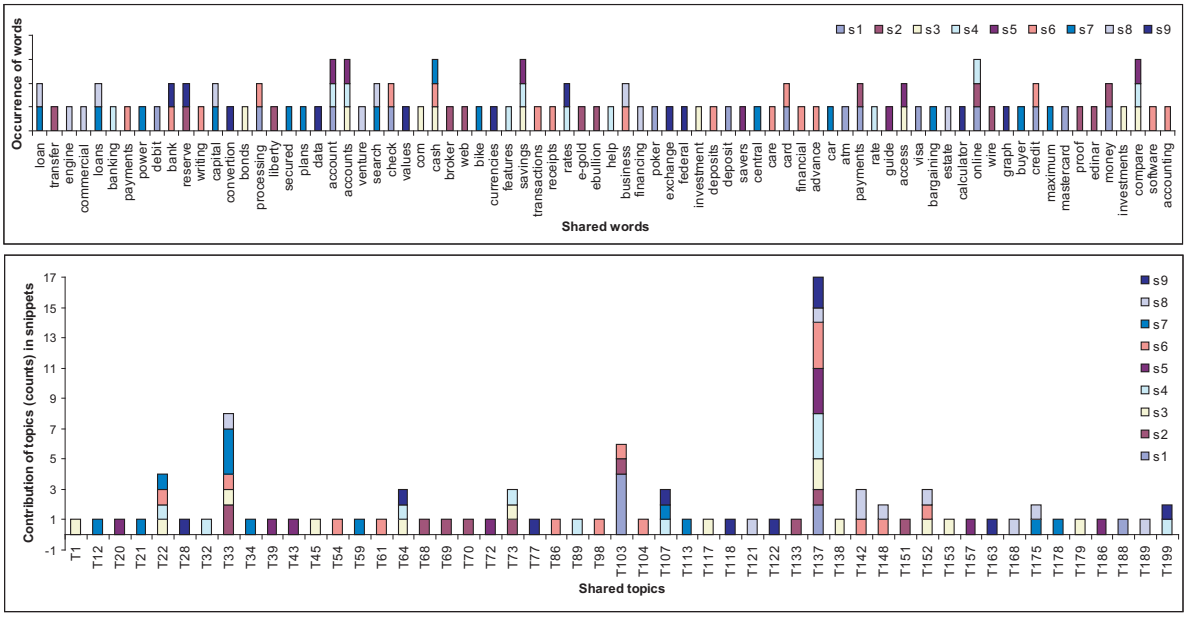
(snippet5) compare savings accounts savings accounts compare savings account savings account savings account guide compare access savers *topic:20* *topic:39* *topic:43* *topic:72* *topic:137* *topic:137* *topic:137* *topic:157* *topic:186*

(snippet6) bank transactions sap business accounting software sap business care financial processing cash receipts check writing deposits advance payments credit card payments *topic:22* *topic:33* *topic:54* *topic:61* *topic:86* *topic:98* *topic:103* *topic:104* *topic:137* *topic:137* *topic:137* *topic:142* *topic:148* *topic:152*

(snippet7) secured loans central capital loans car loan van loan bike loan ll search secured loan plans maximum bargaining power ll cash buyer *topic:12* *topic:21* *topic:22* *topic:33* *topic:33* *topic:33* *topic:34* *topic:59* *topic:107* *topic:113* *topic:175* *topic:178*

(snippet8) search business loan capital business capital search engine business loans venture capital commercial estate financing *topic:33* *topic:121* *topic:137* *topic:142* *topic:142* *topic:148* *topic:152* *topic:168* *topic:175* *topic:189*

(snippet9) exchange rates currencies convertion calculator exchange rates graph values data federal reserve bank *topic:28* *topic:64* *topic:77* *topic:107* *topic:118* *topic:122* *topic:137* *topic:137* *topic:163* *topic:199*



our experiments were trained using the same parameter setting. Those context predicates (words and topics) whose occurrence frequency in the whole training data is smaller than 3 will be eliminated, and those features (a pair of a context predicate and a class label) whose frequency is smaller than 2 will also be cut off. The Gaussian prior over feature weights σ^2 was set to 100.

VII. Building Advertising Matching & Ranking Models with Hidden Topics

A. Topic Inference for Ads & Target Pages

Topics that have high probability $\vartheta_{\underline{m},k}$ will be added to the corresponding Web page/ad $\underline{m}$. Each topic integrated into a Web page/ad will be treated as an *external term* and its frequency is determined by its probability value. Technically, the number of times a topic k is added to a Web page/ad $\underline{m}$ is decided by two parameters *cut-off* and *scale*:

$$Frequency_{\underline{m},k} = \begin{cases} round\left(scale \times \vartheta_{\underline{m},k}\right), \text{if } \vartheta_{\underline{m},k} \geq cut\text{-}off \\ 0, \text{if } \vartheta_{\underline{m},k} < cut\text{-}off \end{cases}$$

where *cut-off* is the topic probability threshold, *scale* is a parameter that determines the topic frequency added.

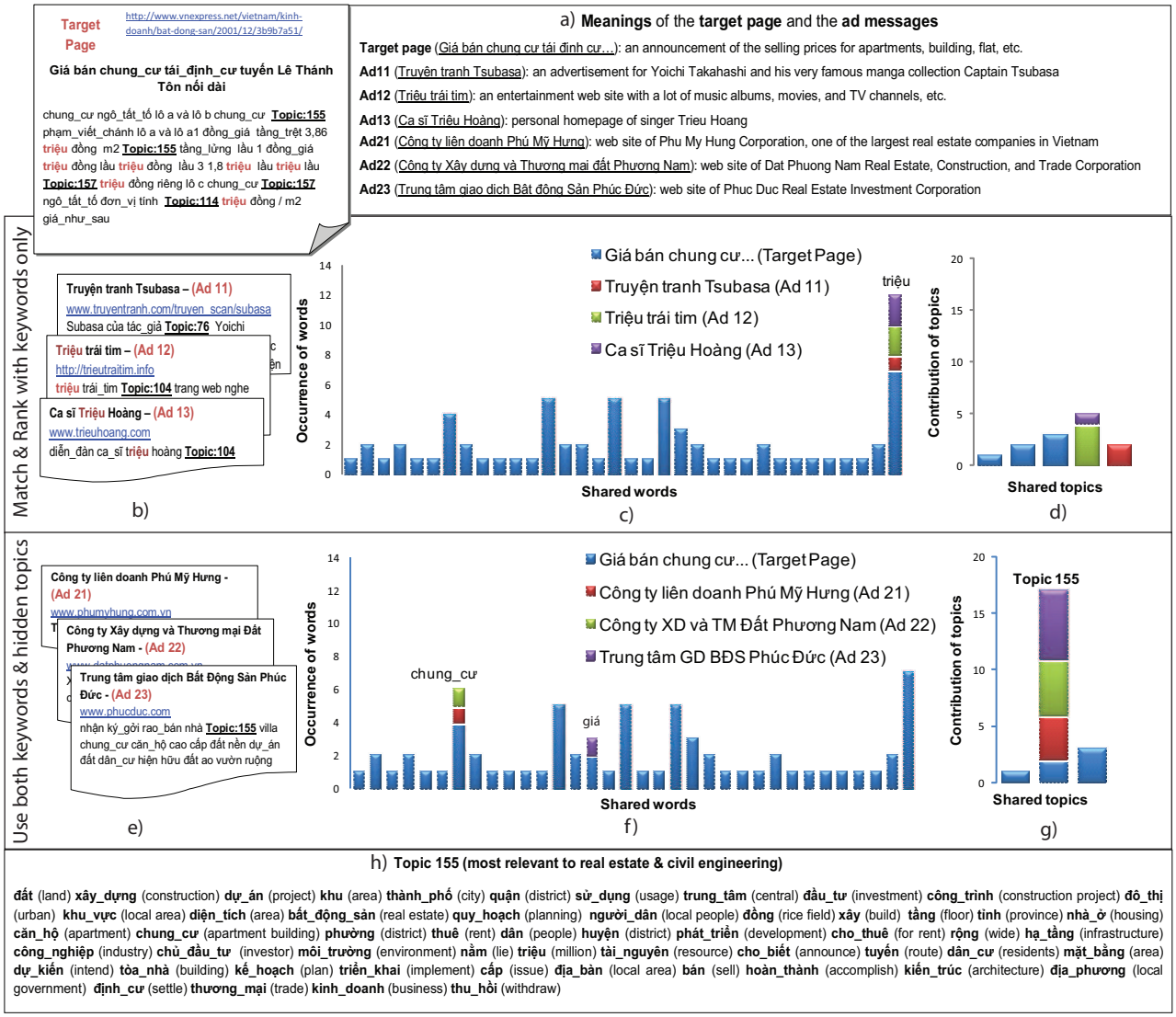
An example of topic integration into ads is illustrated in Figure 7. The ad is about an entertainment Web site with a lot of music albums. After doing topic inference for this ad, hidden topics with high probabilities are added to its content in order to make it enriched and more topic-focused.

B. Matching and Ranking

After being enriched with hidden topics, Web pages and ads will be matched based on their cosine similarity. For each page, ads will be sorted in the order of its similarity to the page. The ultimate ranking function will also take into account the keyword bid information. But this is beyond the scope of this paper.

We verified the contribution of topics in many cases that normal keyword-based matching strategy cannot find appropriate ad messages for the target pages. Since normal matching is based on only the lexical feature of Web pages and ads, it is sometimes deviated by unimportant words which are not practical in matching. An example of such case is illustrated in Figure 8. The word "trieu" (million) is repeated many times in the target page, hence given a high weight in lexical matching. The system then misleads in proposing relevant ad messages for this target page. It puts ad messages having the same high-weighted word "trieu" in the top ranked list (c). However, those ads are totally irrelevant to the target page as the word "trieu" can have other meanings in Vietnamese. The words "chung cu" (apartment) and "gia" (price) shared by top ads proposed by our method (Ad_{21}, Ad_{22}, Ad_{23}) and the target page, on the other hand, are important words although they do not have as high weights as the unimportant word "trieu" (f). However, by analyzing topics for them, we can find out their latent semantic relations and thus realize their relevance since they share the same topic 155 (g) and important words "chung cu" (apartment) and "gia" (price). Topics analyzed for

Fig. 8. A visualization of an example of a page-ad matching and ranking *without* and *with* hidden topics. This figure attempts to show how hidden topics can help improve the matching and ranking performance by providing more semantic relevance between the target Web page and the ad messages. All the target page and the ads are in Vietnamese. The target page is located at the top-left corner. Part (a) explains the meanings of the target page and the ads; Part (b) shows the top three ads (i.e., Ad_{11}, Ad_{12}, and Ad_{13}) in the ranking list *without* using hidden topics (i.e., using keywords only); Part (c) is the visualization of shared words between the target page and the three ads Ad_{11}, Ad_{12}, Ad_{13}; Part (d) visualizes the shared topics between the target page and Ad_{11}, Ad_{12}, Ad_{13}; Part (e) shows the top three ads (i.e., Ad_{21}, Ad_{22}, and Ad_{23}) in the ranking list *using* hidden topics; Part (f) visualizes the shared words between the target page and the three ads Ad_{21}, Ad_{22}, Ad_{23}; Part (g) shows the shared topics between the target page and Ad_{21}, Ad_{22}, Ad_{23}; Part (h) shows the content of hidden topic number 155 (most relevant to *real estate and civil engineering*) that is much shared between the target page and the ads Ad_{21}, Ad_{22}, Ad_{23}.



the target page and each ad message are integrated to their contents as illustrated in Figure 8, b & c.

VIII. Evaluation

So far, we have introduced two general frameworks whose aim is to (1) improve the classification accuracy for short text/Web documents and (2) improve the matching and ranking performance for online contextual advertising. The two frameworks are very similar in that they both rely on hidden topics discovered from huge external text/Web document collections (i.e., universal datasets). In this section, we describe thoroughly two experimental tasks: "Domain Disambiguation for Web Search" and "Contextual Advertising for Vietnamse Web". The first task demonstrates the classification framework and the second demonstrates the contextual matching and ranking framework. To carry out these experiments, we took advantage of the two large text/Web collections Wikipedia and VnExpress News Collection together with their hidden topics that have been presented in Sections V-A and V-B. We will see how the hidden topics can make the data more topic-focused and semantically related in order to solve the earlier mentioned challenges (e.g., sparse data problem and homonym phenomena); and eventually improve the classification and matching/ranking performance.

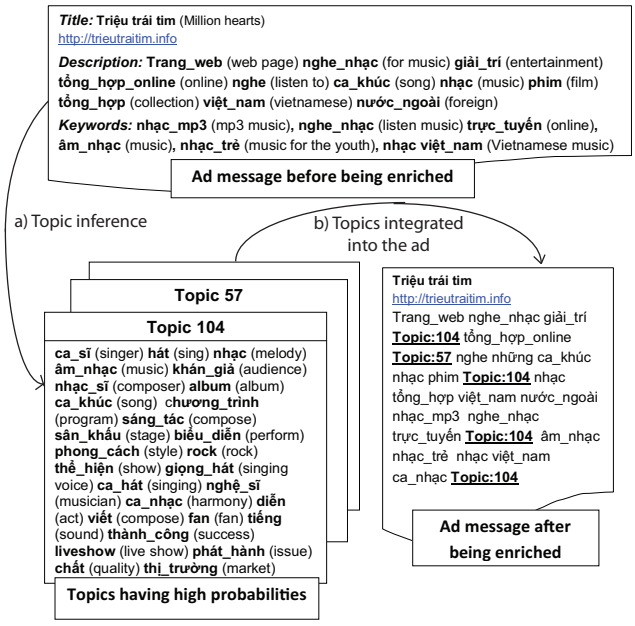
A. Domain Disambiguation for Web Search with Hidden Topics Discovered from the Wikipedia Collection

Clustering Web search results have been an active research topic during the past decade. Many clustering techniques were proposed to place search snippets into topic– or aspect–oriented clusters [41], [42]. This trend has achieved great successes in which Vivisimo is one of the most successful search clustering engines on the Web.

Web search domain disambiguation is different from clustering in that it attempts to put search snippets into one of predefined domains as in Table IV. In this task, hidden topics were discovered from Wikipedia data as described in Section V-A. Both labeled training and testing data were retrieved from Google search using JWebPro[5]. Topic

[5]JWebPro: http://jwebpro.sourceforge.net

Fig. 7. An example of topic integration into an ad message



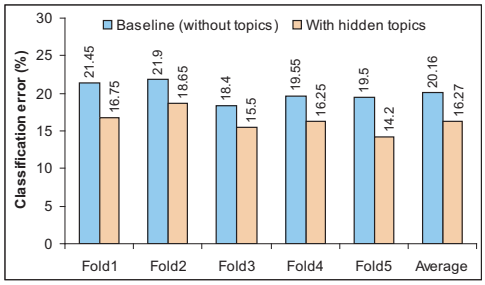
inference for data is as described in Section VI-B and demonstrated in Figure 6. All the classifiers were built using JMaxEnt[6].

TABLE IV
GOOGLE SNIPPETS AS TRAINING & TEST DATA

Search phrases for training & test data are exclusive				
Domain	**Training data**		**Test data**	
	#Phrs.	#Snip.	#Phrs.	#Snip.
Business	60	1,200	10	300
Computers	60	1,200	10	300
Culture-Arts-Ent.	94	1,880	11	330
Education-Science	118	2,360	10	300
Engineering	11	220	5	150
Health	44	880	10	300
Politics-Society	60	1,200	10	300
Sports	56	1,120	10	300
Total		10,060		2,280

1) Experimental Data: To prepare the labeled training and test data, we performed Web search transactions using various phrases belonging to different domains. For each search transaction, we selected the top 20 or 30 snippets from the results to ensure that most of them belong to the same domain. For example, for domain *Business*, we searched 60 phrases and selected the top 20 snippets for each, and got a total of 1,200 snippets. Note that our search phrases for training and test data are *totally exclusive* to make sure that test data is really difficult to classify. The data statistics are shown in Table IV. The training and test data are available online[7].

Fig. 9. 5-fold CV evaluation on the training set



[6]JMaxEnt (in JTextPro): http://jtextpro.sourceforge.net
[7]http://jwebpro.sourceforge.net/data-web-snippets.tar.gz

Fig. 10. Test-out-of-train evaluation with different sizes of labeled data

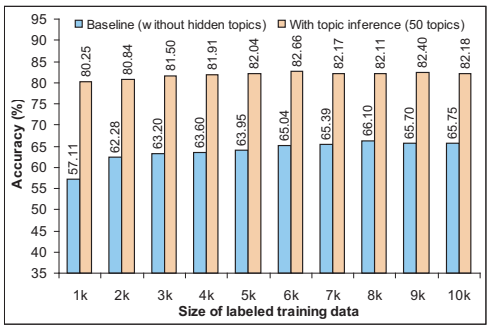
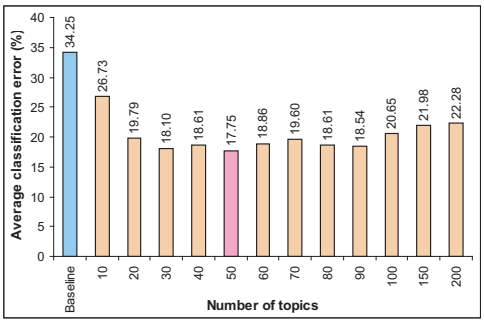


Fig. 11. Classification error reduction changing according to #topics

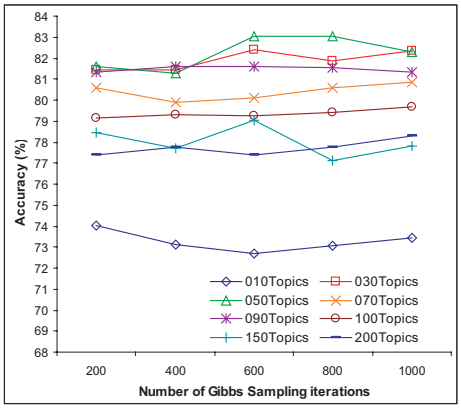


2) Results and Analysis: In order to examine the classification accuracy within the training data, we randomly divided the training set into five equal partitions and performed a five-fold cross validation. For each fold, we ran experiments to measure the classification error of the baseline model (i.e., without hidden topics) and the model that was built according to the framework with 50 Wikipedia topics. The comparison of error is shown in Figure 9. The last two columns show the average error reduction over the five folds. As in the figure, we can reduce the error from 20.16% to 16.27% (removing 19% of error), i.e., increasing the classification accuracy from 79.84% to 83.73%. This means that even within the training data with a certain level of words shared among the snippets, our method is still able to improve the accuracy significantly.

We did a more important experiment by training many classifiers on different sizes of the training data ranging from 1,000 to 10,000 labeled snippets, and measured the accuracy on the test set. Keep in mind that the search phrases for test data and training data are *totally exclusive* so that their snippets share very few common words. This makes the test data really difficult to predict correctly if using traditional classifiers. The results of this experiment are shown in Figure 10. This figure highlights two points. First, the proposed method can achieve an impressive improvement of accuracy when classifying new data, that is, increasing from an accuracy of 65.75% of the baseline to 82.18% (i.e., eliminating more than 52% of error). This means that the method efficiently works with sparse and previously unseen data. Second, we can achieve a high accuracy with even a small amount of labeled training data. When the size of training changes from 1,000 to 10,000 snippets, the accuracy with hidden topics changes slightly from 80.25% to 82.18% (while the baseline accuracy increases nearly 10%, from 57.11% to 65.75%).

The next experiment is to see how the classification accuracy (and error) changes if we change the number of hidden topics of Wikipedia. We estimated many LDA models for the Wikipedia data with different numbers of topics (from 10 to 100, 150 and 200). After doing topic inference, 12 MaxEnt classifiers were built on the training data according to different numbers of topics. All of them, and a baseline classifier, were evaluated on the test data, and the classification error was measured. The change of classification error is depicted in Figure 11. We can see that the error reduces gradually with 10, 20 topics, reduces most around 50 topics, and then increases

Fig. 12. The accuracy changes according to #topics and #Gibbs iterations



gradually. The error changes slightly from 20 to 100 topics. This means that the accuracy is quite stable with respect to #topics.

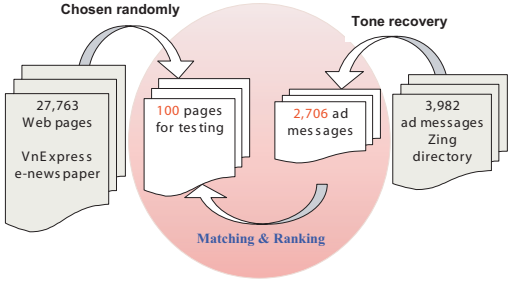
The last experiment with Web search snippets is to examine how Gibbs Sampling influences the classification accuracy. We estimated different LDA models on the Wikipedia data with different numbers of topics (K = 10, 30, . . ., 100, 150, 200). To estimate parameters of each model, we ran 1,000 Gibbs Sampling iterations, and saved the estimated model at every 200 iterations. At these saving points, we performed topic inference for training and test data, building MaxEnt classifiers on the training data, and then measured the accuracy on the test set. The results are shown in Figure 12. As depicted in the figure, for those numbers of topics that give high performance (e.g., 30, 50 70, 90 topics), the accuracy changes slightly with respect to the different numbers of Gibbs Sampling iterations. Although it is hard to control the convergence of Gibbs Sampling, we observed that it is quite fast and yields stable results after the "burn-in" period (about 200 iterations).

B. Contextual Advertising for Vietnamse Web: Matching & Ranking with Hidden Topics from the VnExpress News Collection

In contextual advertising, matching and ranking ad messages based on their relevance to the targeted web page are important factors. As stated earlier, they help increase the likelihood of visits to the website pointed by the ad. In Section III-B and Section VII, we have introduced our framework to perform this task. In this framework, we use hidden topics discovered from a huge external document collection (i.e., the universal dataset) in order to solve the sparse data problem (i.e., few common keywords between target pages and ads) and the synonym & homonym phenomena. The universal dataset is the VnExpress news collection that has been described earlier in Section V-B. All the test target Web pages and the test ads were collected from Vietnamese Web sites. We will present experimental data, experimental settings, evaluation methodology & metrics, as well as the experimental results & analysis in more detail in the following subsections.

1) Experimental Data: We quantified the effect of matching and ranking *without* and *with* hidden topics using a set 100 target Web pages and 2,706 unique ads.

Fig. 13. The test data collection for evaluation



For target Web pages, we chose 100 pages randomly from a set of 27,763 pages crawled from VnExpress, one of the highest ranking e-newspapers in Vietnam. Those pages were chosen from different topics: Food, Shopping, Cosmetics, Mom & children, Real Estate, Stock, Jobs, Law, etc. These topics are primarily classified on the e-newspaper. Note that the information of these classified topics is not used in our experiments, just for reference here only.

For ad messages, as contextual advertising has not yet been applied in Vietnam to our knowledge, it is difficult to find a real Vietnamese advertisement collection. Up to now, advertisement types in Vietnam are mainly banners, thus such kind of real ad messages are not available. We have also contacted some online advertising companies, such as VietAd[8], a company in which keyword-based advertising system has once been tested in Vietnamnet [9]. However, their database was just for testing and the number of such advertisements was only a few (less than 10 ads). In order to conduct the experiments, we chose another resource: Zing.VN[10], a rich online directory of Vietnamese Web sites. It suits the form of contextual ad messages perfectly. Each ad message is composed of four parts: title, Web site's URL, its description, and some important keywords. After crawling all 3,982 ad messages from Zing.VN directory using Nutch[11], we preprocessed the data by doing sentence segmentation, word tokenization, removal of filters and non topic-oriented words. Nevertheless, keywords in this database are almost none-tone, so we cannot use them directly to enhance the matching performance. However, keywords play an important role in contextual advertising. The contribution of them in matching and ranking has been proved through experiments and affirmed in many previous studies [10], [32], [14]. Therefore, we recovered tone for all keywords of the ads in order to improve the performance. After preprocessing, we selected 2,706 unique ads for evaluation. The test data collection that includes 100 target Web pages and 2,706 ads are available online[12] for download.

2) Experimental Settings: In order to evaluate the importance of keywords in contextual match and the contribution of hidden topics in this framework, we performed some different matching strategies as follows: First, to assess the impact of keywords in contextual match, we implemented two retrieval baselines following the approach of Ribeiro-Neto et al. [32]. The first strategy is called AD, that means matching a Web page and an ad message using ad's title and description only. The second is AD_KW, that is, matching a Web page and an ad message using ad's additional keywords, which have already been tone-recovered. The similarity between a target Web page and ads is computed using *cosine* function. Then, the similarity of a Web page p and an ad message a is defined as follows.

$$sim_{AD}(p,a) = similarity(p,a)$$

$$sim_{AD\_KW}(p,a) = similarity(p, a \cup \text{KWs})$$

where KWs is the set of keywords associated with the ad message a. We then used these two settings as the baselines for comparison.

Second, to compare the contribution of hidden topics with additional terms in the Impedance Coupling method [32], we implemented the AAK_EXP method as follows:

$$sim_{AAK\_EXP}(p,a) = similarity(p \oplus r, a \cup \text{KWs})$$

where AAK_EXP follows the implementation in [32], r is the set of additional terms provided by Impedence Coupling technique. These terms are extracted from a large enough dataset of additional web pages. First, the relation between this dataset, its terms and each target web page is represented in a Bayesian network model. Let $\mathcal{N}$ be the set of the k most similar documents d_j to each target page. The probability that term T_i in set $\mathcal{N}$ is a good term for representing a topic of the web page P is then determined as follows:

$$P(T_i|P) = \rho((1-\alpha)w_{i0} + \alpha \sum_{j=1}^{k} w_{ij} sim(r, d_j)) \qquad (7)$$

[8]VietAd (Vietnam Advertisement Company): http://vietad.vn/
[9]Vietnamnet: http://vietnamnet.vn/
[10]Vietnamese Zing Directory: http://directory.zing.vn/
[11]Nutch (an open-source search engine): http://lucene.apache.org/nutch/
[12]Ad data: http://gibbslda.sourceforge.net/ContextualAd-TestData.zip

where ρ is a normalizing constant, w_{i0} and w_{ij} are the weights associated with term T_i in page P and in document d_j. The number of additional terms in r for enriching target page P is decided by the given threshold β.

To perform this method, we used the same 40,268 Web pages in universal dataset as additional dataset. In the experiment, we chose $\beta = 0.05$ as mentioned in [32] and $\alpha = 0.7$, which adjust the amount of additional terms in each target page. The set r will then be integrated with content of each target page to match with advertisements.

TABLE V
EXPERIMENTAL SETTINGS FOR PAGE-AD MATCHING & RANKING

Settings	Target Web page p	Ad message a
AD	p	a
AD_KW	p	$a \cup$ KWs
AAK_EXP	$p \oplus r$	$a \cup$ KWs
HT[m]_[n]	$p \oplus \text{HTs}_p$	$a \cup \text{KWs} \oplus \text{HTs}_a$

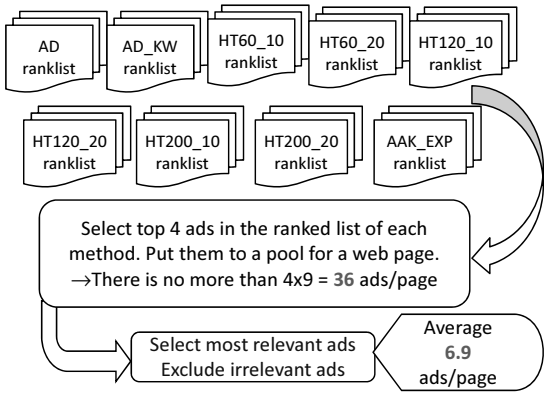
- p = Web page content
- a = ad title + a short ad description
- KWs: tone-recovered keywords from the original ad messages
- r: the set of additional terms in the Impedance Coupling method
- HTs_p, HTs_a: are two sets of most likely hidden topics inferred from the topic model for p and a, respectively
- $\oplus$ means the inclusion of hidden topics by doing topic inference
- m: = 60, 120, 200, the #hidden topics in the topic models
- n: = 10, 20, the scaling value used for hidden topic integration
- We have 3 baselines (AD, AD_KW, AAK_EXP) and 6 hidden topic based HT60_10, HT60_20, HT120_10, HT120_20, HT200_10, HT200_20

In order to evaluate the contribution of hidden topics, we carried out six different experiments, which are called HT (hidden topic) strategies. After doing topic inference for all Web pages and ads, we expanded their vocabularies with their most likely hidden topics. As described earlier in Section VII, each Web page or ad have a distribution over hidden topics. We then chose topics having high probability values to enrich that page or ad. The similarity measure between a target Web page p and an ad a, denoted by $sim_{HT[m]\_[n]}(p, a)$, is computed as follows.

$$sim_{HT[m]\_[n]}(p, a) = similarity(p \oplus \text{HTs}_p, a \cup \text{KWs} \oplus \text{HTs}_a)$$

in which m and n are the total number of topics in the topic model and the scale value as described in Table V, respectively. HTs_p and HTs_a, as explained in Table V, are the two sets of most likely hidden topics inferred from the topic model for p and a, respectively. In the experiments, we used the value cutoff of 0.05 and tried two different scale values: 10 and 20. We therefore performed six experiments: HT60_10, HT60_20, HT120_10, HT120_20, HT200_10, HT200_20.

Fig. 14. Preparation for test ads



3) Evaluation Methodology and Metrics: To evaluate the extent to which hidden topics contribute to the improvement of matching and ranking performance, we prepared the test advertising data for 100 target Web pages with the same methodology used in Ribeiro-Neto et al. [32]. The test data preparation, as depicted in Figure 14, is as follows. First, we started by matching each Web page to all the ad messages and ranking them to their similarities. 9 methods proposed 9 different rank lists of ad messages to a target page. Since the number of ad messages is large, these lists can be different from this method to another with little or no overlap. To determine the precision of each method and compare them, we selected top four ranked ads of each method and put them into a pool for each target page. Consequently, each pool will have no more than 36 ad messages. We then selected from these pools the most relevant ads and excluded irrelevant ones. On average, each Web page will be matched with 6.9 ads eventually. To calculate the precision of each method, we used 11-point average score [29], a metric often used for ranking evaluation in IR.

4) Results and Analysis: We used the method AD_KW as a baseline for our experiments which uses hidden topics. We examined the contribution of hidden topics using different estimated models: the model of 60, 120 and 200 topics. As illustrated in Figure 15 and Table VI, using hidden topics significantly improves the performance of the whole framework. Figure 15 shows seven precision-recall curves of seven experiments in which the most inner line is the baseline and all the others are with hidden topics. From these curves, we can see the extent to which hidden topics can improve matching and ranking accuracy, and how the parameter values (i.e., number of topics, scale value) affect the performance. From Table VI, we can see hidden topics help increase the precision on average from 66% to 73% and reduces almost 21% error (HT200_20).

Fig. 15. Precision-recall curves of the baseline (without hidden topics) and the 6 settings with hidden topics

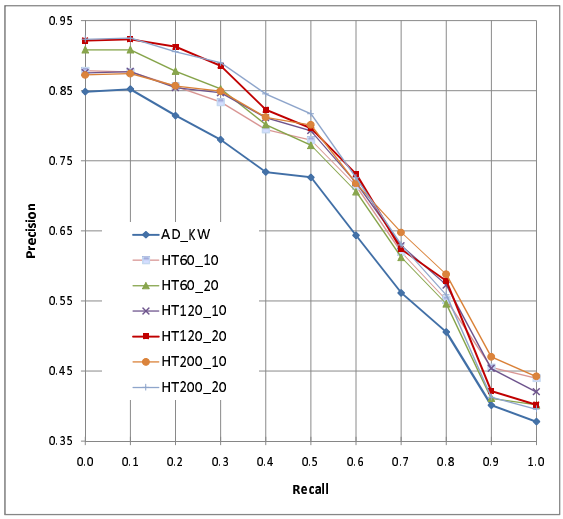


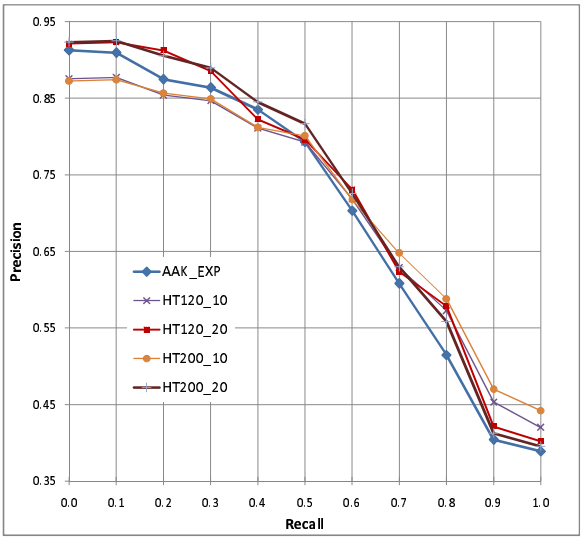
TABLE VI
PRECISIONS OF POSITIONS #1, #2, #3 AND 11-POINT AVERAGE

Methods	Correct ads found				11-point avg. precision
	#1	#2	#3	Totals	
AD	72	56	54	182	**50.22%**
AD_KW	79	71	66	216	**65.85%**
AAK_EXP	87	76	79	242	**70.98%**
HT60_10	82	78	74	233	70.89%
HT60_20	85	80	74	239	70.87%
HT120_10	82	78	79	239	71.37%
HT120_20	87	83	76	246	**72.88%**
HT200_10	81	78	78	237	72.09%
HT200_20	88	79	81	248	**72.97%**

For the overall methods, we also calculated the number of corrected ads found in the first, second and third position of the rank lists proposed by each strategy (#1, #2, #3 in Table VI). Because in contextual advertising, normally, we only consider some first ranked ads, we want to examine the precision of these top slots. It also reflects that the precision of our hidden-topic methods is higher than that of the baseline matching method. Moreover, the precision at position 1 (#1) is generally higher than that of position 2 and 3 (#2, #3). If the system ranks the relevant ads near the top of the ranking

list, it is possible that the system can suggest most appropriate ads for the corresponding page. It therefore shows the effectiveness of the ranking system.

Fig. 16. Precision-recall curves of the Impedance Coupling method and the Hidden Topics method



Impedance Coupling method is another solution to match Web pages and ads by expanding the text of the web page, which is similar to the Hidden Topic idea in reducing vocabulary impedance. To compare with this method, we use the same web pages in universal dataset to extract additional terms. As shown in Figure 16, the accuracy of AAK_EXP method is almost the same as HT60 method but less than HT120 and HT200 method (Table VI). However, one limitation of the Impedance Coupling method is time consuming. Using the same number of web pages in universal dataset, for each target page, the system has to compute the similarity of the target page with each document in the dataset to find the relation with k most similar pages. After that, for every terms in this set, the probability that this term is good for enriching the target page is calculated to find the set of best terms r. This process takes a considerable computational time while the number of web pages and ads in real application is very large. For Hidden Topic method, although estimating the universal dataset would take a long time, once it is estimated, the model can be used for topic inference for web pages and ads. This process is very fast and only takes several seconds to do topic inference for thousands of short documents. This is the main advantage of Hidden Topic method in comparison with Impedance Coupling.

Finally, we also quantified the effect of the number of topics and its added amount to each Web page and ad by testing with different topic models and adjusting the scale values. As indicated in Table VI, the performance of 120 and 200-topic models yields a better result than 60-topic model. However, there is no considerable change between 120-topic and 200-topic models, also in the quantities of added topics to each page and ad. It can therefore conclude that the number of topics should be large enough to discriminate the difference of terms to better analyze topics for Web pages and ads. However, when the number of topics is large enough, the performance of the overall system becomes more stable.

The framework has shown its efficiency through a variety of experiments against the basic method using syntactic information only and the method adding terms from additional web pages. In practice, the results record an error reduction of 21% in the method using 200-topic model over the normal matching strategy without hidden topics. This indicates that this high quality contextual advertising framework is easy to implement and practical in reality.

IX. Conclusions

We have presented a general framework to build classification and matching/ranking models for short and sparse text/Web data by taking advantage of hidden topics from large-scale external data collections. The framework mainly focuses on several major problems we might have when processing such kind of data: data sparseness and synonym/homonym problems. Our approach provides a way to make sparse documents more related and topic-focused by performing topic inference for them with a rich source of global information about words/terms and concepts/topics coming from universal datasets. The integration of hidden topics helps uncover and highlight underlying themes of the short and sparse documents, helping us overcome difficulties like synonyms, hyponyms, vocabulary mismatch, noisy words for better classification, clustering, matching, and ranking. In addition to sparseness and ambiguity reduction, a classifier or matcher built on top of this framework can handle future data better as it inherits a lot of unknown words from the universal dataset. Also, the framework is general and flexible to be applied to different languages and application domains. We have carried out two careful experiments for two evaluation tasks and they have empirically shown how our framework can overcome data sparseness and ambiguity in order to enhance classification, matching, and ranking performance.

The future studies will be focusing on improving the framework in a number of ways: how to estimate and adjust the number of hidden topics automatically; find more fine-grained topic analysis, e.g. hierarchical or nested topics, to meet more sophisticated data and applications; pay more attention to the consistency between the universal dataset and the data we need to work with; and incorporate keyword bid information into ad ranking to achieve a full solution to matching and ranking for online contextual advertising.

Acknowledgment

This work is fully supported by the research grant No.P06366 from Japan Society for Promotion of Science (JSPS). We would also like to say special thanks to the Editor-in-Chief, the Associate Editor, and the anonymous reviewers for reviewing our manuscript and giving us a lot of useful comments and suggestions.

References

[1] L. Baker and A. McCallum. Distributional clustering of words for text classification. In *ACM SIGIR*, 1998.
[2] P. Baldi, P. Frasconi, and P. Smyth. *Modeling the Internet & the Web: probabilistic methods & algorithms*. Wiley, 2003.
[3] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *ACM SIGIR*, 2007.
[4] A. Berger, A. Pietra, and J. Pietra. A maximum entropy approach to natural language processing. *Comp. Ling.*, vol.22, no.1, pp.39–71, 1996.
[5] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text cate. *JMLR*, vol.3, pp.1183–1208, 2003.
[6] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *JMLR*, vol.3, pp.993–1022, 2003.
[7] D. Blei and J. Lafferty. A correlated topic model of *Science*. *The Annals of Applied Statistics*, vol.1, no.1, pp.17-35, 2007.
[8] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using Web search engines. In *WWW*, 2007.
[9] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co–training. In *COLT*, 1998.
[10] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *ACM SIGIR*, 2007.
[11] L. Cai and T. Hofmann. Text categorization by boosting automatically extracted concepts. In *ACM SIGIR*, 2003.
[12] J. Cai, W. Lee, and Y. Teh. Improving WSD using topic features. In *EMNLP-CoNLL*, 2007.
[13] P. Chatterjee, D. Hoffman, and T. Novak. Modeling the clickstream: Implications for Web-based advertising efforts. *Marketing Science*, vol.22, no.4, pp.520–541, 2003.
[14] M. Ciaramita, V. Murdock, and V. Plachouras. Semantic associations for contextual advertising. *Journal of Electronic Commerce Research*, vol.9, no.1, pp.1–15, 2008.

[15] S. Deerwester, G. Furnas, and T. Landauer. Indexing by latent semantic analysis. *Journal of the American Society for Info. Science*, vol.41, no.6, pp.391–407, 1990.
[16] L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *ACM SIGIR Forum*, 2006.
[17] I. Dhillon and D. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learn.*, vol.29, no.2-3, pp.103–130, 2001.
[18] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.
[19] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE PAMI*, vol.6, pp.721–741, 1984.
[20] T. Griffiths and M. Steyvers. Finding scientific topics. *The National Academy of Sciences*, vol.101, pp.5228–5235, 2004.
[21] IAB: Interactive Advertising Bureau. IAB Internet advertising revenue report. *Technical Report*, 2008.
[22] T. Joachims. Text categorization with SVMs: learning with many relevant features. In *ECML*, 1998.
[23] G. Heinrich. Parameter estimation for text analysis. *TR*, 2005.
[24] T. Hofmann. Probabilistic LSA. In *UAI*, 1999.
[25] T. Hofmann. Latent semantic models for collaborative filtering. *ACM TOIS*, vol.22, no.1, pp.89–115, 2004.
[26] A. Lacerda, M. Cristo, M. Andre, G. Fan, N. Ziviani, and B. Ribeiro-Neto. Learning to advertise. In *ACM SIGIR*, 2006.
[27] T. A. Letsche and M. W. Berry. Large-scale information retrieval with latent semantic indexing. *Information Science*, 100(1-4):105–137, 1997.
[28] D. Liu and J. Nocedal. On the limited memory BFGS method for large-scale optimization. *Math. Programming*, vol.45, pp.503–528, 1989.
[29] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to information retrieval*. Cambridge University Press, 2008.
[30] D. Metzler, S. Dumais, and C. Meek. Similarity measures for short segments of text. In *ECIR*, 2007.
[31] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. In *WWW*, 2008.
[32] B. Ribeiro-Neto, M. Cristo, P. Golgher, and E. Moura. Impedance coupling in content-targeted ad. In *ACM SIGIR*, 2005.
[33] M. Sahami and T. Heilman. A Web-based kernel function for measuring the similarity of short text snippets. In *WWW*, 2006.
[34] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
[35] P. Schonhofen. Identifying document topics using the Wikipedia category network. In *IEEE/WIC/ACM Web Intelligence*, 2006.
[36] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, vol.34, no.1, pp.1–47, 2002.
[37] R. Wang, P. Zhang, and M. Eredita. Understanding consumers attitude toward advertising. In *AMCIS*, 2002.
[38] X. Wei and W. Croft. LDA-based document models for ad-hoc retrieval. In *ACM SIGIR*, 2006.
[39] W. Yih, J. Goodman, and V. Carvalho. Finding advertising keywords on Web pages. In *WWW*, 2006.
[40] W. Yih and C. Meek Improving similarity measures for short segments of text. In *AAAI*, 2007.
[41] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to Web search results. In *WWW*, 1999.
[42] H. Zeng, Q. He, Z. Chen, W. Ma, and J. Ma. Learning to cluster Web search results. In *ACM SIGIR*, 2004.

Xuan-Hieu Phan received his BS and MS degrees in information technology from College of Technology, Vietnam National University, Hanoi in 2001 and 2003, respectively. He then earned his PhD degree in information science from Japan Advanced Institute of Science and Technology in 2006. He was a postdoctoral fellow of Japan Society for Promotion of Science (JSPS) at Graduate School of Information Sciences, Tohoku University from 2006 to 2008. He is currently a research fellow at Centre for Health Informatics, University of New South Wales. His research interests include natural language processing, machine learning, information retrieval, Web&text mining, and business intelligence.

Cam-Tu Nguyen received her BS and MS degree in information technology from College of Technology, Vietnam National University, Hanoi in 2005 and 2008, respectively. She is now a PhD candidate at Graduate School of Information Sciences, Tohoku University. Her research interests include natural language processing, text/Web data mining, and multimedia information retrieval.

Dieu-Thu Le received her BS in information technology from College of Technology, Vietnam National University, Hanoi in 2008. She is now one of the master students of European Masters Program in Language and Communication Technology (LCT). Her main research interests include natural language processing, information retrieval, and online advertising & business intelligence.

Le-Minh Nguyen received the BS degree in information technology from Hanoi University of Science, and the MS degree in information technology from Vietnam National University, Hanoi in 1998 and 2001, respectively. He received the PhD in information science from Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST) in 2004. He is now an assistant professor at Graduate School of Information Science, JAIST. His research interests include text summarization, machine translation, natural language processing, machine learning, and information retrieval.

Susumu Horiguchi received the BEng, MEng, and PhD degrees from Tohoku University in 1976, 1978, and 1981, respectively. He is currently a professor and the chair of the Department of Computer Science, Graduate School of Information Science, and the chair of the Department of Information Engineering, Faculty of Engineering, Tohoku University. He was a visiting scientist at the IBM T.J. Watson Research Center from 1986 to 1987. He was also a professor in Japan Advanced Institute of Science and Technology (JAIST). He has been involved in organizing international workshops and conferences sponsored by the IEEE, IEICE, IASTED, and IPS. He has published more than 150 technical papers on optical networks, interconnection networks, parallel algorithms, high-performance computer architectures, VLSI/WSI architectures, and data mining. He is a senior member of the IEEE and member of IPS and IASTED.

Quang-Thuy Ha received his BS degree in computation and mathematics from Hanoi University of Sciences (HUS) in 1978 and his PhD degree in Information Technology in 1997 from HUS, Vietnam National University, Hanoi. He is currently an associate professor in information systems and serving as a vice rector of College of Technology (Coltech), Vietnam National University, Hanoi. He is also the head of the Knowledge Engineering and Human-Computer Interaction Laboratory at Coltech. His main research interests include rough sets, data mining and knowledge engineering, and information retrieval.