International Journal of Cognitive Informatics and Natural Intelligence
Volume 15 • Issue 4 • October-December 2021

A Classification Framework of Identifying Major Documents With Search Engine Suggestions and Unsupervised Subtopic Clustering

Chen Zhao, University of Tsukuba, Japan
Takehito Utsuro, University of Tsukuba, Japan
Yasuhide Kawada, Logworks Co., Ltd., Japan

ABSTRACT

This paper addresses the problem of automatic recognition of out-of-topic documents from a small set of similar documents that are expected to be on some common topic. The objective is to remove documents of noise from a set. A topic model-based classification framework is proposed for the task of discovering out-of-topic documents. This paper introduces a new concept of annotated {\ it search engine suggests}, where this paper takes whichever search queries were used to search for a page as representations of content in that page. This paper adopted word embedding to create distributed representation of words and documents and perform similarity comparison on search engine suggestions. It is shown that search engine suggestions can be highly accurate semantic representations of textual content and demonstrate that the document analysis algorithm using such representation for relevance measure gives satisfactory performance in terms of in-topic content filtering compared to the baseline technique of topic probability ranking.

KEYWORDS

Document Processing, Embedding, Search Engine Suggests, Subtopic, Text Mining, Topic Model, Unsupervised Learning, Word2vec

INTRODUCTION

Topic models are statistical models in text analysis and are functionally capable of discovering hidden semantic structures from documents. It estimates a probability distribution of topics on documents and is commonly applied to document clustering (Xie & Xing, 2013). They are associated with various text mining applications as effective tools for clustering large amounts of unstructured documents. This paper studies the LDA topic model (Blei, Ng, & Jordan, 2003), where input documents are assigned probability distributions of a fixed number of latent topics. Such distributions are estimated through Gibbs sampling on raw words in each document. The topic model infers both the probability of word membership to each topic and the probability of topic membership to each document so that every document receives distinct lists of probability weights corresponding to latent topics. According to

DOI: 10.4018/IJCINI.20211001.oa42

such a weight list, the most likely topic a document statistically belongs to can be easily identified at the index of probability maximum. However, a common problem of probabilistic topic models is that topics are often erroneously inferred because an incoherent set of documents may get assigned to the same topic. Namely, the topic model allocates the same maximum topic likelihood to irrelevant documents. Topic clusters generally contain documents with similar words. For this reason, more effective document analysis mechanism is useful for classifying whether a document is irrelevant to its allocated topic. In this paper, the authors define, on top of topic models, the notion of *major documents* as documents that semantically belong to the topic they are allocated to. Conversely if a document is an outlier from its allocated topic, it is defined as *minor documents,* or *noise documents* that are not considered appropriate candidates for that topic.

The primary task of this paper is to design a unified framework that discriminates minor documents from major ones. The authors first collected all-Japanese Web pages about four major categories which they call a *query focus* in this work: "job hunting", "marriage", "hay fever" and "apartment". All these query focuses are closely associated to trending topics among Japanese Internet communities. For a single query focus, collected pages are separately applied to the topic model and then hard clustered based on the maximum probability topic assigned to documents. The design of proposed algorithm is based upon two observable facts. First, a topic cluster commonly consists of documents on diverse content that can further be divided into subtopic groups, where these subtopics are still interpretable from human perspective. Second, major documents are semantically close to the majority of others in the same cluster. In other words, subtopics of major documents are more likely to be shared by other in-topic candidates. According to these facts, the proposed algorithm requests similarity comparison between pairs of documents within each topic. The authors also introduce a new concept of *annotated suggestions*, those query suggestions proposed by search engines. A query suggestion is said to belong to a page if searching with suggestion leads to that page. Alternatively, they are also called *search engine suggestions* in this paper. Word embedding is adopted to create distributed representation of words and documents, to enable similarity comparison among query suggestions. The primary reason of expecting search engine suggestions to accurately represent context semantics is that these *suggestions* are collected from search engine history logs reflecting user interest and behavior. Following frequently searched queries more likely leads to pages with useful content that tends to be trusted by users. This assumption is verified by experiments.

The proposed framework incorporates three unsupervised models of learning document features. Those models differ in training data and embedding techniques. For all the models, the same proposed algorithm can be applied concerning feature similarity. Features learned by different models provides distinct classification results. Evaluation discrepancy in term of precisions/recalls vary as well. The outputs are evaluated against reliable ground truth that are manually labeled.

Experiments demonstrate that topic model probability does not secure absolute document relevance in real use cases. Henceforth additional techniques are desirable for document clustering of better quality and this brings forward motivation of this research. One apparent advantage of the proposed framework, as the major contribution of this paper, is that input documents directly serve as training data and absolutely no supervised training is involved. No expensive labor is thus required for data labeling. This unsupervised nature implies practical significance in many real-world applications, wherever noise reduction mechanism is needed to filter out unwanted contents, such as search engine improvement, low-cost training data generation for text categorization, content-filtering based information retrieval, etc. The rest of this paper follows organization below. First it briefly introduces the structure of experimental datasets and how the LDA topic model is applied, followed by the proposed generic algorithm of automatic subtopic labeling based on vector similarity comparison. Next it illustrates in detail how the unsupervised models are trained to obtain respective document representations. A baseline approach that selects major documents by topic probability ranking is described as well. Then it explains some guidelines to abide by when evaluation labels

are manually produced, along with explanation on how average precisions and recalls are computed. Evaluation of experiments are shown next before conclusion.

RELATED WORK

The document classification framework proposed in this paper gains insight from existing topic modeling metrics. Related works mainly cover previous study on topic models and information extraction surrounding search engine functionality. Blei et al.(2003) proposed perplexity as a topic quality metric. de Waal and Barnard (2008) discussed the problem of vocabulary dependence in perplexity and raised the concept *topic stability* as an evaluation technique. Similarly there are other works (Wallach, Murray, Salakhutdinov, & Mimno, 2009; Mimno & Blei, 2011) studying correctness of topic modeling. This paper differs from those works by focusing on in-topic contents instead of overall probabilistic behavior of the topic model.

As for in-topic content evaluation, there also exist quite a few quality evaluation schemes that measure how meaningful topics are. Those related works include Chang, Boyd-Graber, Gerrish, Wang, and Blei (2009) using manual evaluation on held-out keywords and other works (Newman, Karimi, & Cavedon, 2009; Newman, Lau, Grieser, & Baldwin, 2010; Musat, Velcin, Trausan-Matu, & Rizoiu, 2011; Stevens, Kegelmeyer, Andrzejewski, & Buttler, 2012; Chan & Akoglu, 2013; Aletras & Stevenson, 2013) that rely on external knowledge resources such as Wikipedia, Wordnet[10], search engine information and news corpora. Lau, Newman, and Baldwin (2014) and Röder, Both, and Hinneburg (2015) did some systematic comparison among various topic evaluation techniques. AlSumait, Bardara, Gentle, and Domeniconi (2009) recognized junk topics through unsupervised analysis so that topics will be ranked by semantic legitimacy. Compared to aforementioned techniques, this paper not only performs topic evaluation but also directly filters noise documents thus effectively refining a topic by improving semantic coherence in an intuitive way. The expected outcome is more coherent and natural topic modeling consisting of less garbage and hence textual data aggregation benefits from better quality.

User generated queries and search engine results provide valuable inspiration to modern information retrieval. Existing works studying search result aggregation also involve techniques other than what is proposed in this paper such as alternative topic models (Harashima & Kurohashi, 2010), keyword resolution (Toda & Kataoka, 2005; Shibata, Bamba, Shinzato, & Kurohashi, 2009) and concept clustering (Leung, Ng, & Lee, 2008). This paper differs from similar works by incorporating word embedding into search engine *suggestions* for topic model improvement.

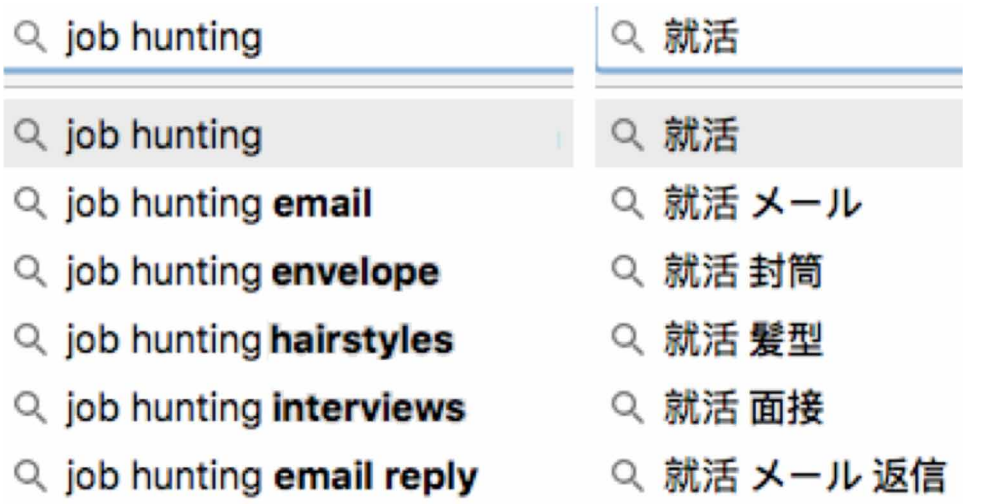
COLLECTING SUGGESTIONS FROM SEARCH ENGINE AND WEB PAGES

Collecting Query Suggestions

For every given query focus keyword, the authors specify about 100 types of Japanese hiragana characters to Google from which they collect not exceeding 1000 suggestion keywords. For example, when "就活 あ" ("job hunting a") is typed into the search field, a keyword list is popped up starting with the character "a" such as "aisatsu" and "anata no tsuyomi". This paper defines these suggested tokens after the query focus word as search engine suggestions.

In this example, the search engine provides suggestions "aisatsu" and "anata no tsuyomi", both of which will be collected. All such suggestions of a single query focus constitute set S. Figure 1 shows an example on how the search engine provides various suggested keywords around the example query focus keyword "job hunting" ("shyu-katsu" in the Japanese case). These keywords are indication of frequently queried items and trending topics over the recent time span of collecting them because they are typically user generated content.

Figure 1. An Example of Query Suggestions Provided by the Search Engine



Collecting Web Pages

Using the suggestions collected in the previous section combined with the query focus keyword as input queries (in the form of AND search), the authors always collect the first 20 pages returned per query at the time of crawling the search engine.[1] The set of Web pages queried by suggested keyword s can be represented as $D(s,N)$ where N is 20 as a constant standing for the top N pages.

Every Web page is annotated with suggestions that were used for querying that page. As different suggestions could lead to identical Web pages, one page could consequently possess multiple suggestions. So it is necessary to maintain a set of suggestions $S(d)$ for every Web page d so that $S(d)$ contains all its suggested keywords. Suppose the search engine lists d in its top N feedbacks when crawled with some suggestion in $S(d)$. Suggestion keywords of a Web page are saved as follows.

$$S(d)=\{s\in S|d\in D(s,N)\}$$

THE LDA TOPIC MODEL

This paper employs LDA (Latent Dirichlet Allocation) (Blei et al., 2003; Bolelli, Ertekin, & Giles, 2009) to model topic distributions among documents. Given a preset constant K representing the number of output topics, the LDA topic model takes a collection of documents and estimates the word distribution $p(w|z_n)(w\in V)$ where V is the vocabulary set [2] for every topic $z_n\,(n=1,2,\ldots,K)$. Every document also gets assigned a topic distribution $p(z_n|d)$. This paper adopts GibbsLDA++ [3] as the toolkit while the parameters are predefined through preliminary evaluation by tuning the number of topics K. In this paper, $K=60$ for the query focus "marriage" and 50 for all the others throughout all the experiments.

Let D be the document set as input data and K be the number of topics. The topic model performs soft clustering on documents by assigning topic distribution $p(z_n|d)$ for every $d(d \in D)$. For application in this paper, only the most probable topic is used for clustering documents in D. Then every document *d* is assigned to the topic index of the maximum probability among all its $p(z_n|d)$ to materialize soft cluster estimation into hard clusters. The following formula defines how a hard topic cluster is created.

$$D(z_n) = \left\{ d \in D | z_n = argmax_{z_u (u=1,2,\ldots,k)} P(z_u \mid d) \right\}$$

The overall effect is that for every topic z_n, there exists a unique set of corresponding documents belonging to z_n. As hard clustering is enforced, a document being assigned multiple topics is strictly prevented, so there exist no overlap among topic clusters $D(z_n)$ for $n = 1, 2, \ldots, K$. Moreover, since every document in $D(z_n)$ gets assigned to topic n subject to the condition that $p(z_n|d)$ maximizes probability out of K topics. Within every topic cluster $D(z_n)$, it is feasible to sort documents by $p(z_n|d)$ in descending order so that every topic cluster is now an ordered document list. One straightforward interpretation of a topic cluster being a list is that the list starts with the most likely document for this topic as $p(z_n|d)$ decreases along the list order. Strictly maintaining this order is a crucial step for experiments illustrated in following sections.

SELECTING MAJOR DOCUMENTS BY SUBTOPIC CLUSTERING

Before starting to design classification algorithm on major documents versus minor documents, preliminary investigation was conducted about some unique patterns of minor documents in each topic. The main discovery is that content of minor documents is rarely in common with major documents. From human perspective, authors of this paper define such content distinction as *subtopic* because documents within the same topic cluster can still cover diverse subtopics. The above investigation brings forward a conjecture that subtopics of minor documents tend to have fewer occurrences inside their topic cluster and the opposite case goes for major documents. The proposed classification algorithm below assumes that given a topic cluster $D(z_n)$, the subtopics of minor documents are prone to scarcity that tends to isolate them from major documents. Therefore the algorithm first attempts to recover the subtopic of every document in $D(z_n)$ and then counts the occurrence of these subtopics. Documents of high occurrence subtopics are classified as major ones. Since inter-document similarity is a valuable reference to quantify how many documents in $D(z_n)$ share a similar subtopic of $d \in D(z_n)$, a general similarity based algorithm is implemented in order to discover subtopics in $D(z_n)$.

Given a topic cluster $D(z_n)$, for every possible pair of different documents d and *d'*, the algorithm computes the similarity between their features in the form of vectors notated as $v(d)$ and $v(d')$. Given a similarity lower bound θ_{lbd}, any document pair d and d' with similarity greater or equal to θ_{lbd} is selected as a candidate pair. As a result, the following set of candidate document pairs $D_p(z_n, \theta_{lbd})$ is generated from topic cluster $D(z_n)$.

$$D_p\left(z_n,\theta_{lbd}\right)=\left\{\left(d,d'\right)|d,d'\in D\left(z_n\right),d\neq d',sim\left(v\left(d\right),v\left(d'\right)\geq\theta_{lbd}\right)\right\}$$

Notice that pair order is not considered for $D_p\left(z_n,\theta_{lbd}\right)$, i.e., $\left(d,d'\right)$ is equivalent to $\left(d',d\right)$. Documents of a candidate pair are designated to have the same subtopic. Then, given a similarity lower bound θ_{lbd}, the topic cluster $D\left(z_n\right)$ of documents for the topic z_n is decomposed into disjoint subtopic clusters $D_{sub}^1\left(z_n,\theta_{lbd}\right),\ldots,D_{sub}^k\left(z_n,\theta_{lbd}\right)$:

Here, subtopic clusters $D_{sub}^1\left(z_n,\theta_{lbd}\right),\ldots,D_{sub}^k\left(z_n,\theta_{lbd}\right)$ are to satisfy the following two requirements.

i. In each subtopic cluster $D_{sub}^i\left(z_n,\theta_{lbd}\right)\left(i=1,\ldots,k\right)$, for each document $d\left(d\in D_{sub}^i\left(z_n,\theta_{lbd}\right)\right)$, at least one other document within the subtopic cluster $D_{sub}^i\left(z_n,\theta_{lbd}\right)$ holds similarity above or equal to the similarity lower bound θ_{lbd} with d.
ii. No two documents from different subtopic clusters hold similarity above or equal to the similarity lower bound θ_{lbd}.

$D_{sub}^i\left(z_n,\theta_{lbd}\right)\left(i=1,\ldots,k\right)$ in practice stands for the set of documents in $D\left(z_n\right)$ estimated to be of the same subtopic.[4] Eventually, documents in subtopic clusters of greater cardinality are selected as major documents, represented as the set below.

$$major\left(z_n,\theta_{lbd}\right)=\bigcup_{D_{sub}^i\left(z_n,\theta_{lbd}\right)\geq d_f}D_{sub}^i\left(z_n,\theta_{lbd}\right)$$

In this paper, d_f is a constant of 3 so that this algorithm selects every document with an estimated subtopic of at least 3 occurrences in $D\left(z_n\right)$ as major documents.[5] Subtopics containing fewer than 3 documents are treated as noise documents due to scarcity in all the experiments mentioned in this paper.

DOCUMENT SIMILARITY MEASURES

The previous section demonstrates the algorithm of generating subtopic clusters based on similarity measures among document features assuming every document has available features. In this section, three different techniques of document features are explained in detail. These techniques mainly involve two popular unsupervised models of distributed representation of words and documents, *word2vec* model (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), or *word embedding* and *doc2vec* (Le & Mikolov, 2014) which is a generalization of *word2vec* and is also capable of directly training sentence and paragraph vectors.

Word2vec Based Measure

Suggestion Frequency and Word2vec

Since multiple suggestions could coexist as queries to the same Web page as mentioned in Section *Collecting Web Pages*, it is useful to select the most properly suggested word from $S\left(d\right)$ in case of

multiple candidate suggestions so that the selected word gives a best description of content in d. For the purpose of experimental tasks this paper defines the notion of suggestion frequency $f\left(s, z_n\right)$. For suggestion s, its frequency in topic z_n is defined as the occurrence count of documents containing s in $D\left(z_n\right)$.

$$f\left(s, z_n\right) = \left|\left\{d \in D\left(z_n\right) | s \in S\left(d\right)\right\}\right|$$

As suggestion frequency $f\left(s, z_n\right)$ for s is defined within the scope of specific topic z_n, it is possible for the same suggestion s to have varying frequencies depending on the given topic z_n. This research extracts the suggestion of the highest in-topic frequency from $S\left(d\right)$ to represent document d and denotes it as $s_{max}\left(d\right)$ as below.

$$s_{max}\left(d\right) = \left\{s \in S\left(d\right) | s = argmax_s f\left(s, z_n\right), d \in D\left(z_n\right)\right\}$$

Word embedding is then applied to train the feature vector of $s_{max}\left(d\right)$ representing d. The intuition behind in-topic suggestion frequency is that higher occurrence of documents $d \in D\left(z_n\right)$ containing s indicate that s is a dominant keyword within the scope of topic z_n and is more likely to provide a correct indication of page content compared to lower frequency suggestions. This finding stems from the property of the LDA topic model that assigns similar documents to the same topic with higher probability. Because the set $S\left(d\right)$ of every document d is conserved even after the topic model is applied, suggestions of documents in a single topic can be very useful indication on how diverse documents in this topic are. For example, if some suggestion is shared by the majority of documents in $D\left(z_n\right)$ it is very likely to represent a common subtopic in $D\left(z_n\right)$.

The first approach to train the feature vector $v\left(d\right)$ of documents attempts to train a word2vec[6] model (Mikolov et al., 2013) so that the most frequent suggestion $s_{max}\left(d\right)$ can be embedded into semantically meaningful vector and used as a vector to represent document d. This paper adopts the

skip-gram model from one of the two optional flavors in word2vec which takes target words from the training text and learns to predict context words from the target. It is an unsupervised vector space model that maps semantically similar words to proximity in the embedded vector space. In principle, a neural probabilistic language model (Bengio, Schwenk, Senécal, Morin, & Gauvain, 2006) is trained to maximize the likelihood $p\left(w_t, c\right)$ of every word in the training data given previous words (or *history*) c for word w_t, where the compatibility of word w_t with c can be evaluated as a dot product. The objective is likelihood of the entire training data.

$$L^{SG} = \sum_{t=1}^{T} \sum_{c \in C_{w_t}} \log p(w_t \mid c)$$

The compatibility of context word c with w_t is $p(w_t \mid c)$ by softmax probability.

$$p(w_t \mid c) = \frac{\exp\left(v_c \cdot \tilde{v}_{w_t}\right)}{\sum_{w' \in V} \exp\left(v_c \cdot \tilde{v}_{w'}\right)}$$

Computing the log likelihood becomes unacceptably expensive given a large vocabulary given large vocabulary, so the skip-gram model in practice uses noise-contrastive estimation which only maximizes the target word likelihood meanwhile minimizing random sampled noise word likelihood, subject to the noise-contrastive estimation (Dyer, 2014).

$$L^{SG} = \sum_{t=1}^{T} \sum_{c \in C_{w_t}} \left\{ \log \sigma\left(v_c \cdot \tilde{v}_{w_t}\right) + \sum_{k=1}^{K} \log \sigma\left(-v_c \cdot \tilde{v}_{\substack{\emptyset \\ w}}\right) \right\}$$

Given some context h from the training data, log likelihood of every word w_t is optimized against its contrastive words w. In case of K negative samples, the expectation of negative probability distribution is approximated as Monte Carlo average over K. A trained word2vec model represents a word as an embedding vector of preset dimension. In case of representing a document d with a vector $v\left(s_{max}\left(d\right)\right)$ of a suggestion keyword $s_{max}\left(d\right)$, similarity between features $v\left(d\right)$ of documents d are calculated as the cosine similarity of the feature vector $v\left(s_{max}\left(d\right)\right)$ of $s_{max}\left(d\right)$:

$$sim\left(v\left(d\right), v\left(d'\right)\right) = \frac{v\left(s_{max}\left(d\right)\right) \cdot v\left(s_{max}\left(d'\right)\right)}{v\left(s_{max}\left(d\right)\right) v\left(s_{max}\left(d'\right)\right)}$$

The word2vec model trained in this paper uses an embedding size of 256 and minimum count of 5, so that words of occurrence below 5 will be ignored. The skip-gram context window of sampling target words is constantly 5 for every experiment in this paper.

Japanese Version of Wikipedia as The Training Data

As the skip-gram model is scales very well to large training data, the first experiment starts with entry text from Wikipedia containing all available articles in Wikipedia.[7] Then it extracts the embedding vector $v\left(s_{max}\left(d\right)\right)$ of suggestion $s_{max}\left(d\right)$ as representation of document d. After training is completed, the trained model vocabulary is looked up for vector representation $v\left(s_{max}\left(d\right)\right)$ for every document d. One problem for such annotation is that not every suggestion keyword will be successfully embedded into vector representation since some of the words are never present throughout Wikipedia texts during training phase. This leads to direct consequence that some documents fail to get valid features $v\left(s_{max}\left(d\right)\right)$ due to lack of embedding of $s_{max}\left(d\right)$. Accordingly the proposed algorithm is forced to ignore documents without valid $v\left(s_{max}\left(d\right)\right)$ at the run time. Table 1 lists a complete overview of dataset used for training and evaluation. The second row lists the total number of distinct search engine suggestions with respect to individual query focuses. In case of Wikipedia entry text as training data, numbers of successfully embedded suggested keywords (also called *embedded suggestions*) is listed in the third row. For example, 709 out of 959 collected suggested

keywords of the query focus "marriage" possess vector representation from the trained model due to occurrence above the minimum count 5.

Japanese Version of Wikipedia as The Training Data

As it now becomes clear that even a fine trained word2vec model lacks embedded suggestion keywords that are crucial to training vector representation of documents, the experiment launched a second attempt to mitigate this problem by integrating the Wikipedia text and collected Web pages as training data. Table 1 lists the numbers of Web pages appended to Wikipedia entry text per query focus. For example, 13,256 pages in the form of roughly 60MB text files for the query focus "marriage" is used along with Wikipedia text as training data. Appending the query focus text to the original Wikipedia data ensures that the combined version of training data contains a lot more words in query suggestions thus producing much better vocabulary in the trained word2vec model because these documents are Web pages collected through the annotated suggestion word as is described in Section *Collecting Web Pages*. The probability of Web pages containing keywords used for fetching them from the search engine is reasonably much higher than random Wikipedia entry articles.

As expected many more suggestion keywords are successfully recovered. In the case of "marriage" as an example, Table 1 shows that 771 out of 959 suggestion keywords are now embedded using the improved training data and the same set of training parameters. However, it is not yet able to embed every suggestion because a Web page d does not explicitly guarantee coverage of $s_{max}(d)$ in its context.

Doc2vec Based Measure

In addition to word2vec based models working on annotated suggestions, the authors develop one more approach that directly learns paragraph vectors without concern with any suggestion. doc2vec[8] model (Le & Mikolov, 2014) is a generalization of word2vec model that performs identical training mechanism to word2vec. In addition to training vectors for words, word2vec also models vector $p(d)$ for individual paragraphs in the input corpus. This paper utilizes the Distributed Memory Model of Paragraph Vectors (PV-DM). The PV-DM model appends a paragraph ID at the time of target word probability estimation given some context, so that the paragraph ID is equivalent to another word from the perspective of the softmax classifier. As training data, every d is regulated to be a single paragraph and directly take the paragraph vector as the feature of d as $v(d) = p(d)$. In this way, doc2vec trains a valid $v(d)$ for every document in the query focus. To train doc2vec models all the Web pages collected with regard to a query focus are used as training data without Wikipedia text. Numbers of pages in each query focus used for training word2vec are displayed as total Web pages in Table 1. For example, 13,256 input paragraphs were used as input when word2vec trains features $p(d)$ of documents belonging to "marriage".

In this paper, training parameters for doc2vec follow empirical evaluation (Lau & Baldwin, 2016) that was researched in preexisting experiments related to doc2vec, where embedding size is 300 and the

window size is 5. The minimum count is selected as 1 so that every word in the training text is considered without dropping any low frequency word. Upon completion of training the doc2vec model, similarity between features $v(d)$ of documents d using doc2vec are calculated as the cosine similarity of $p(d)$ in a similar way to Formula (2).

$$sim\left(v(d), v(d')\right) = \frac{p(d) \cdot p(d')}{p(d)\, p(d')}$$

Table 1. Complete Dataset Overview

Query Focus	marriage	job hunting	hay fever	apartment
# suggestion keywords	959	926	850	958
# embedded suggestions above minimum count (Wikipedia only)	709	559	632	684
# embedded suggestions above minimum count (Wikipedia + Web pages)	771	671	695	758
# total Web pages	13,256	12,078	9,745	13,742

EVALUATION

Evaluation Procedures and Metrics

This paper picks up 10 topic clusters from each of the 4 query focuses composing a total evaluation dataset of 40 topics. Within every topic, the top 30 documents are selected by topic probability ranking $p\left(z_n|d\right)$ in descending order specified in Section *THE LDA TOPIC MODEL*. Since one query focus contains 300 labeled documents, the total evaluation set consists of altogether 1,200 documents from 4 query focuses as is listed in Table 1. Since word2vec based measures do not guarantee valid representation for every document, similarity comparison of $v\left(s_{max}\left(d\right)\right)$ is not available for these documents.[9] The selection algorithm described in Section *Selecting Major Documents by Subtopic Clustering* ignores such documents for candidate pair selection in Formula (1). Still, all the documents are evaluated. Those not positively classified, $d \notin major\left(z_n, \theta_{lbd}\right)$, are considered minor documents. This implies that documents without valid $v\left(d\right)$ are never selected as major ones by the algorithm.

Evaluation of major/minor document classification is given in terms of precision/recall calculated from prediction set $major\left(z_n, \theta_{lbd}\right)$ and reference set $labeled\left(z_n\right)$ (ground truth) for topic z_n. The reference set is created by independent human judgement and never interfered with by the proposed algorithm. As for human judgement, $d \in D\left(z_n\right)$ is labelled as a major document if and only if there exist at least 2 other documents of the same subtopic in $D\left(z_n\right)$. Equivalently, human judgement considers $d \in D\left(z_n\right)$ as a major document if the subtopic of d appears at least 3 times in $D\left(z_n\right)$. Practically, reference set $labeled\left(z_n\right)$ consists of true major documents with manually labeled subtopics occurring at least 3 times in the topic cluster.

Figure 2 illustrates a concrete example of such evaluation on a real topic cluster, where the word2vec based similarity measure is employed with Japanese version of Wikipedia along with Web pages as the training data (the similarity lower bound $\theta_{lbd} = 0.32$). A majority of documents in this example topic are articles sharing experience of successful communication through proper self-analysis and personal statement in Japanese style on-site job interviews. Based on reference labels, there are two major subtopics that cover skills of making self-analysis and stating personal strength and weakness. For similarity metrics, false negative occurs when a reference major document gets classified as a minor one due to isolation of its suggestion keyword. In this example, the document with suggestion "catch phrase" is a false negative from prediction as this word does not hold enough similarity with suggestion keyword of major documents such as "self-introduction", even if it in fact covers the same

subtopic of self-analysis. The documents with keyword "about yourself" and "weakness'" are predicted to be the same subtopic due to similar vector representation of original Japanese tokens. False positives result from high suggestion similarity of reference minor documents. For instance, suggestion keyword "frustration" is close to "weakness" in terms of word embedding but articles on negative job-hunting experience and has little relevance to constructive interview techniques.

For individual topics, precision is defined as the proportion of correctly selected major documents out of all selected documents with Formula (3). Recall for z_n is defined as the proportion of selected major documents out of all the true major documents in the reference set with Formula (4).

$$precision\left(z_n,\theta_{lbd}\right)=\frac{major\left(z_n,\theta_{lbd}\right)\cap labeled\left(z_n\right)}{major\left(z_n,\theta_{lbd}\right)}$$

$$recall\left(z_n,\theta_{lbd}\right)=\frac{major\left(z_n,\theta_{lbd}\right)\cap labeled\left(z_n\right)}{label\left(z_n\right)}$$

Moreover, precision and recall across topic clusters are calculated in macro average and micro average. The macro average is the mean of precisions and recalls of all topics while the micro average takes the sum of correctly selected major documents from all topics and calculates the proportions respectively for micro precision and micro recall. Formula (5) to Formula (8) give formal definition on macro/micro precision/recall.

$$precision_{macro}\left(\theta_{lbd}\right)=\frac{1}{K}precision\left(z_n,\theta_{lbd}\right)$$

$$recall_{macro}\left(\theta_{lbd}\right)=\frac{1}{K}recall\left(z_n,\theta_{lbd}\right)$$

$$precision_{macro}\left(\theta_{lbd}\right)=\frac{\sum_{n=1}^{K}major\left(z_n,\theta_{lbd}\right)\cap labeled\left(z_n\right)}{\sum_{n=1}^{K}major\left(z_n,\theta_{lbd}\right)}$$

$$recall_{macro}\left(\theta_{lbd}\right)=\frac{\sum_{n=1}^{K}major\left(z_n,\theta_{lbd}\right)\cap labeled\left(z_n\right)}{\sum_{n=1}^{K}labeled\left(z_n\right)}$$

The number of topic clusters K varies by query focuses, with $K=60$ for "marriage" and 50 for the rest as stated in Section *THE LDA TOPIC MODEL*. For every query focus, macro/micro precisions and recalls are separately evaluated for different similarity lower bound values θ_{lbd} on the interval $\left[-1.0,1.0\right]$ which is domain for cosine similarity, with stepwise increment of 0.02. One θ_{lbd} value corresponds to a macro precision/recall pair and a micro precision/recall pair since θ_{lbd} is the only required hyper-parameter for this evaluation. The final evaluation outcome consists of all such precision/recall pairs and plots the correlation between precisions and recalls. Both macro and micro averages are included.

A baseline technique is also evaluated to verify effectiveness of the proposed algorithm in terms of major/minor document classification. The baseline technique unconditionally trusts the topic model

outputs by classifying the top r documents from every topic cluster into major candidates $major\left(z_n, \theta_{lbd}\right)$ expressed in Formula (9), with r being an input parameter.

$$major\left(z_n, r\right) = \{d \in D\left(z_n\right) | rankofp\left(z_n | d\right) \geq r\} \left(r = 1, \ldots, 30\right)$$

The document ranking follows descending order by topic probability $p\left(z_n | d\right)$ mentioned in Section *THE LDA TOPIC MODEL*. The evaluation process starts with r till $r = 30$ and calculates the micro/macro average across topic clusters in the same way as is defined in Formula (5) to Formula (8), with the input hyper-parameter θ_{lbd} replaced with the ranking parameter r.

Figure 3 to Figure 10 display the smoothed version of the above evaluation into 11 precision/recall points. Evaluation outcomes are plot in separate figures based on query focuses. Evaluation on each query focus is presented as a pair of figures depicting micro and macro precision/recall averages.

The baseline technique introduced above is indicated as "Topic Ranking (Baseline)" in Figure 3 to Figure 10.

Discussion on Evaluation Outcomes

Up till now, three different types of document features have been applied to the clustering algorithm explained in Section *Selecting Major Documents by Subtopic Clustering*. Throughout all the four query focuses, suggestion based features give the best overall improvement upon precision within

reasonable range of recall compared to the baseline and the doc2vec based feature. The greatest advantage of word2vec based features is its effective lower bound on lowest recall. It is practically common that major documents in a topic cluster share the same designated suggestion $s_{max}\left(d\right)$ leading to identical features among such documents. In this case their mutual feature similarity is always 1.0, so these documents will be correctly classified for sure regardless of input similarity lower bounds as θ_{lbd} does not exceed 1.0. This pattern contributes to the minimum recall of word2vec based evaluation, compared to doc2vec where recall close to 0 exists. Also notice that word2vec based evaluations do not cover a recall range up to 1.0. This is caused by no-feature documents. In this case, when a major document lacks valid feature $v\left(s_{max}\left(d\right)\right)$, it is never recovered by the proposed technique as a major document regardless of similarity lower bounds, making 1.0 recall impossible.

Two types of the features trained using word2vec model differ in terms of their input training data as mentioned in Section *word2vec based measure* thus leading to slightly different evaluations. Evaluation figures show that combining Wikipedia text and query focus Web pages contributes to the overall evaluation due to an increasing number of valid suggestions getting successful embedding vectors.

However, a few exceptions occur, as can be seen from evaluation figures. For the query focus "marriage", macro precisions of word2vec model trained with Wikipedia text and Web pages are outperformed by the word2vec model trained with Wikipedia text only within some small range of macro recall from 0.5 to 0.6 or so (Figure 4). This is caused by a few topics in "marriage" where the word2vec model trained with Wikipedia text and Web pages is particularly incompetent, dragging down the mean value of all topics' precisions. Similar phenomenon also exists in the query focus "apartment", where micro precisions of word2vec model of combined training data get surpassed by word2vec trained with Wikipedia text only within some range of micro recall from 0.4 to 0.6 (Figure 9). These phenomena result from some interesting property of evaluation dataset. Two major factors are discussed below.

The first factor is that in case a topic cluster consists of a high number of noise documents with no valid suggestion embedding $v\left(s_{max}\left(d\right)\right)$, such noise never participate in similarity measures. Equivalently they are guaranteed never to be positively classified in error. This reduces the number of false positive documents thus contributing to higher precision. The second factor is that at relatively high similarity thresholds, some topics could easily reach 1.0 precision due to 0 false positive. Under such circumstance, topics with a larger number of true positive documents contribute to higher micro precision. There do exist topics where word2vec of only Wikipedia data achieves more true positives than the word2vec of composite data. This is reasonable considering there could exist the same number of major documents with valid suggestion embedding in a topic for both word2vec models.

As for evaluation of doc2vec based model, it generally outperforms the designated baseline technique in terms of micro precision, even though such prevalence is in relatively much smaller margin compared to word2vec. One experimental conclusion from the doc2vec model is that neural based representations like doc2vec do not necessarily secure an outcome advantageous enough over traditional topic models for tasks like topic inference. To be specific, to cluster Web page documents into K topics at granularity around $K = 50 \sim 60$, traditional topic models are still possibly worthwhile options compared to modern fancy approaches in terms of overall performance/cost ratio.

CONCLUSION

The objective of this paper document classification between major and minor candidates inside topic clusters to improve topic coherence by noise filtering. The primary work of this paper is to propose a framework of subtopic clustering algorithm based upon topic model outputs. The main idea is to train unsupervised models for learning document features. It presents three unsupervised approaches to acquire distributed representation of documents relying on query suggestions. Two of them embed suggested words with word2vec using different training data and the other is the doc2vec model directly learning vector representation of documents as paragraphs. Evaluation reveals that the proposed algorithm exhibits sensitivity towards document features in use. Evaluation also indicates that suggestion based word2vec models tend to more easily achieve higher precision while preserving a much better recall than doc2vec and the baseline, due to the reason stated in Section *Discussion on evaluation outcomes*. Although doc2vec model ensures vector representation to every document it does not seem to be the best option for the specific task in this paper. Hence it comes to conclusion that by taking advantage of query suggestions and word embedding, existing topic models can be improved for much better topic production in terms of semantic coherence. Future work includes ensemble models derived from three types of features and more advanced techniques of text understanding such as feature extraction with deeper neural architecture.

REFERENCES

Aletras, N., & Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics* (pp. 13-22). Academic Press.

AlSumait, L., Bardara, D., Gentle, J., & Domeniconi, C. (2009). Topic significance ranking of LDA generative models. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (pp. 67-82). doi:10.1007/978-3-642-04180-8_22

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural probabilistic language models. In D. E. Holmes & L. C. Jain (Eds.), *Innovations in Machine Learning: Theory and Applications* (pp. 137–186). Springer. doi:10.1007/3-540-33486-6_6

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 993–1022.

Bolelli, L., Ertekin, C., & Giles, C. L. (2009). Topic and trend detection in text collections using latent dirichlet allocation. In *Proceedings of the 31st European Conference on Information Retrieval* (pp. 776-780). doi:10.1007/978-3-642-00958-7_84

Chan, H., & Akoglu, L. (2013). External evaluation of topic models: A graph mining approach. In *Proceedings of the 13th IEEE International Conference on Data Mining* (pp. 973-978). doi:10.1109/ICDM.2013.112

Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., & Blei, D. (2009). Reading tea leaves: How humans interpret topic models. Advances in Neural Information Processing Systems, 22, 288-296.

de Waal, A., & Barnard, E. (2008). Evaluating topic models with stability. In *Proceedings of the 19th Annual Symposium of the Pattern Recognition Association of South Africa* (pp. 79-84). Academic Press.

Dyer, C. (2014). Notes on noise contrastive estimation and negative sampling. *Computing Research Repository*.

Harashima, J., & Kurohashi, S. (2010). Summarizing search results using PLSI. In *Proceedings of the 2nd Workshop on Natural Language Processing Challenges in the Information Explosion Era* (pp. 12-20). Academic Press.

Lau, J. H., & Baldwin, T. (2016). An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In *Proceedings of the 1st Workshop on Representation Learning for Natural Language Processing* (pp. 78-86). doi:10.18653/v1/W16-1609

Lau, J. H., Newman, D., & Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th European Chapter of the Association for Computational Linguistics* (pp. 530-539). doi:10.3115/v1/E14-1056

Le, Q. V., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning* (pp. 85-104). Academic Press.

Leung, K. W.-T., Ng, W., & Lee, D. L. (2008). Personalized concept-based clustering of search engine queries. *IEEE Transactions on Knowledge and Data Engineering*, *20*(11), 1505–1518. doi:10.1109/TKDE.2008.84

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems, 26, 3111-3119.

Mimno, D., & Blei, D. (2011). Bayesian checking for topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 227-237). Academic Press.

Musat, C. C., Velcin, J., Trausan-Matu, S., & Rizoiu, M. A. (2011). Improving topic evaluation using conceptual knowledge. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence* (pp. 1866-1871). Academic Press.

Newman, D., Karimi, S., & Cavedon, L. (2009). External evaluation of topic models. In *Proceedings of the 14th Australasian Document Computing Symposium* (pp. 11-18). Academic Press.

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Proceeding of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 100-108). Academic Press.

Röder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining* (pp. 399-408). doi:10.1145/2684822.2685324

Shibata, T., Bamba, Y., Shinzato, K., & Kurohashi, S. (2009). Web information organization using keyword distillation based clustering. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* (pp. 325-330). doi:10.1109/WI-IAT.2009.57

Stevens, K., Kegelmeyer, P., Andrzejewski, D., & Buttler, D. (2012). Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 952-961). Academic Press.

Toda, H., & Kataoka, R. (2005). A search result clustering method using informatively named entities. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management* (pp. 81-86). doi:10.1145/1097047.1097063

Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning* (pp. 1105-1112). Academic Press.

Xie, P., & Xing, E. P. (2013). Integrating document clustering and topic modeling. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence* (pp. 694-703). Academic Press.

ENDNOTES

1. Google Custom Search API was used (https://developers.google.com/custom-search) to scrape Web pages from the search engine.
2. In this paper, as the set V of vocabulary, the set of entry titles of the Japanese version of Wikipedia was used, the version downloaded in March 2014 with about 1,407,000 entries.
3. http://gibbslda.sourceforge.net/
4. Subtopic clusters $\mathrm{D}_{\mathrm{sub}}^{\mathrm{i}}\left(\mathrm{z}_{\mathrm{n}},,_{\mathrm{lbd}}\right)$ are generated through single-linkage agglomerative clustering algorithm subject to the similarity lower bound $,_{\mathrm{lbd}}$ starting with the members of $\mathrm{D}_{\mathrm{p}}\left(\mathrm{z}_{\mathrm{n}},,_{\mathrm{lbd}}\right)$ as initial document pairs.
5. In this paper, only the top 30 documents in every topic is considered for classification. So using $\mathrm{d}_{\mathrm{f}}=3$ to quantitatively define the condition of being major documents becomes a heuristic method for this research.
6. https://radimrehurek.com/gensim/models/word2vec.html
7. This paper uses the Japanese version of Wikipedia updated by February, 2016 containing about 1 million entry pages with a total size of roughly 2.8GB
8. https://radimrehurek.com/gensim/models/doc2vec.html
9. Numbers of documents with valid $v\left(s_{max}\left(d\right)\right)$ in evaluation dataset by query focus are 259, 211, 242, 217 for “marriage”, “job hunting”, “hay fever” and “apartment” for $s_{max}\left(d\right)$ lookup from word2vec model trained with Wikipedia text alone and 268, 257, 264, 240 from the model trained with appended dataset, respectively.