1L3-3in1

Identifying Major Documents with Search Engine Suggests by Unsupervised Subtopic Labeling

Chen Zhao[*1] Yi Ding[*1] Tian Nie[*1*2] Jiaqi Li[*1] Takehito Utsuro[*3]
Yasuhide Kawada[*4]

[*1]Grad. School of Systems and Information Engineering, University of Tsukuba
[*2]Pioneer Corporation
[*3]Faculty of Engineering, Information and Systems, University of Tsukuba
[*4]Logworks Co., Ltd

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. We started with querying the search engine with volumes of Web pages on topics of the same focused area and collected them into clusters of topics by topic modeling. Then we implemented several different techniques to identify in each topic cluster documents that are irrelevant from that topic, respectively. The main difficulty of this problem lies in the relatively small cluster size and high variance of Web page contents. All of our techniques are intuitively based on similarity comparison between documents. Instead of atomically comparing word tokens we adopted word embedding upon which three different document selection models were built. Each model learns some distinct vector representation of documents in unsupervised approaches and this work demonstrates that our document analysis algorithm using such representation for relevance measure gives satisfactory performance in terms of in-topic information filtering.

1. Introduction

Topic models are widely deployed statistical models in text analysis and are functionally capable of discovering hidden semantic structures from documents. Intuitively documents are assigned probability distributions of different topics from which the most likely topic a document belongs to can be inferred. However, such topic assignment is not always accurate in practice as there might exist many documents that do not cover a coherent topic even if they are allocated to the same topic with maximum likelihood by the topic model. In such a case, an effective document selection mechanism is useful for identifying which of these documents are apparently irrelevant from the focus of current topic. In this paper we define the notion of *major documents* as documents that are closely related to the topic they are allocated to. Conversely we define documents out of focus from the topic as *minor documents* that are not considered appropriate candidates for that topic.

The primary task of this paper is to design a unified framework that distinguishes major documents from minor ones on per-topic basis. We collected all-Japanese Web pages on 2 categories we call *query focus* in this work: "job hunting (syukatsu)" and "marriage (kekkon)". For each query focus, the topic model is separately applied to the collected pages that are then hard clustered into 50 topics based on the most likely topic assigned to documents. Our algorithm is designed based on fundamental observation that major documents are semantically close to the majority in its topic. In other words, the subtopics of major documents are more likely to be shared by other in-topic documents. Since the algorithm requests similarity comparison between all possible pairs of documents in the topic, we performed word embedding to create distributed representation of documents. This work incorporates three measures of learning document representations with varying training data to embedding techniques. The algorithm outputs are evaluated against pre-labelled manual dataset for precision and recall calculation.

The rest of this paper follows organization below. First it briefly introduces the structure of experimental datasets along with the LDA topic model, followed by our generic algorithm of automatic subtopic labeling based on vector similarity comparison. Then it illustrates in detail how the three unsupervised models are trained to learn document representation. Evaluation of our experiments are shown next and finally it comes to conclusion.

連絡先: 趙 辰，筑波大学大学院システム情報工学研究科，
〒 305-8573 茨城県つくば市天王台 1-1-1, 029-853-5427

2. Collecting Search Engine Suggests and Web Pages

2.1 Collecting search engine suggests

For a given query focus keyword, we specify about 100 types of Japanese hiragana characters to the search engine from which we then collect up to 1,000 suggests. For example, once we type "就活　あ" ("job hunting" "a") into the search field, a list of suggests are popped out all starting with the character "a" such as "aisatsu" and "anata no tsuyomi". All such suggests of one query focus constitutes the set S.

2.2 Collecting Web pages

Using the Web search engine suggests we collected in Section 2.1 combined with the query focus keyword as queries (in the form of AND search), we always collect the first 20 pages returned per query. The set of Web pages queried by suggest s can be represented as $D(s, N)$ where N is 20 as a constant standing for the top N pages. As previously mentioned we save the search engine suggests for every Web page. Since different search engine suggests could lead to the same Web page, one single Web page could have multiple suggests. So we maintain a suggest set $S(d)$ for each Web page d, so that $S(d)$ contains all the suggests that were used to search the page d. Therefore suggests of a Web page are saved as follows.

$$S(d) \quad = \quad \left\{ s \in S \middle| d \in D(s, N) \right\}$$

3. The LDA Topic Model

This paper employs LDA (Latent Dirichlet Allocation) [Blei03] 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 $p(w \mid z_n)$ $(w \in V)$ where V is the vocabulary set*1 for every topic z_n $(n = 1, \ldots, K)$. Every document also gets assigned a topic distribution $p(z_n \mid d)$. This paper adopts GibbsLDA++*2 as the toolkit while the parameters are tuned through a preliminary evaluation by examining the number of topics as $K = 50$ for all the query focuses in our experiments.

Let D be the document set as input data and K be the number of topics. The topic model estimates soft clustering on documents by assigning topic distribution $p(z_n \mid d)$ for every d $(d \in D)$. We then assign every document d the topic with the maximum probability among all its $p(z_n \mid d)$ to turn the topic model output into hard topic clusters. The following formula defines this process.

$$D(z_n) \quad = \quad \left\{ d \in D \,\middle|\, z_n = \underset{z_u\ (u=1,\ldots,K)}{\text{argmax}} \; P(z_u|d) \right\}$$

The overall effect is that for every topic z_n, there is a unique set of corresponding documents belonging to z_n. Since we comply with hard clustering and never assign multiple topics to the same document, there is no overlap between topic clusters $D(z_n)$ for $n = 1, \ldots, K$.

4. Selecting Major Documents by Unsupervised Subtopic Labeling

Our document selection algorithm assumes that within a topic cluster, contents covered by major documents tend to be more commonly shared than minor documents. In this paper such content distinction between documents in the same topic is called subtopics. The algorithm first attempts to emulate subtopics of each document then counts the occurrence of these subtopics. Documents of high occurrence subtopics are selected as major ones.

To start with, given a topic cluster $D(z_n)$, for every pair of documents d and d', the algorithm computes the similarity between their distributed representations, which are notated as vectors $v(d)$ and $v(d')$. Given a similarity lower bound θ_{lbd}, any document pair d and d' with similarity above θ_{lbd} is selected as a candidate pair. As a result, the following set of candidate document pairs $D_p(z_n)$ is generated from topic cluster $D(z_n)$.

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

Pair order is not considered in $D_p(z_n)$, i.e., (d', d) is equivalent to (d, d'). Documents of a candidate pair are believed to have the same subtopic. The above set of pairs can be efficiently recomputed using union-find algorithm into disjoint sets of documents with each set representing documents of a common subtopic. The subtopic set $D_s(d)$ affiliated with d. Consequently every document $d \in D(z_n)$ gets some set assignment.

$$D_s(d, \theta_{lbd}) \quad = \quad \left\{ d' \middle| (d, d') \in D_p(z_n, \theta_{lbd}) \right\} \;\cup\; \{d\}$$

Eventually, documents in subtopic sets of greater cardinality are selected as major documents, represented as the set below.

$$major\ doc(z_n, \theta_{lbd}) \quad = \quad \left\{ d \in D(z_n) \middle| \; \left\| D_s(d, \theta_{lbd}) \right\| \geq d_f \right\}$$

$D_s(d, \theta_{lbd})$ in practice stands for the set of documents in $D(z_n)$ estimated to be of the same subtopic as d including d itself. In this paper, d_f is a constant of 3 so that this algorithm selects every document of estimated subtopic occurring at least d_f times in $D(z_n)$ as major documents.

5. Document Similarity Measures

This section explains three different ways of learning distributed representation of documents in unsupervised patterns. The main technique involves the Word2Vec model [Mikolov13] which is also called *word embedding* and in recent years has become a prevailing option of learning vector representation of words; and Doc2Vec [Le14] which is a generalization of Word2Vec and is capable of directly learning sentence and paragraph vectors.

5.1 Word2Vec based Measure

5.1.1 Suggest Frequency

For the purpose of our task, we extract the suggest of the highest in-topic frequency from $S(d)$ in order to represent document d, where frequency of a suggest $f(s, z_n)$ is defined as the occurrence of documents in $D(z_n)$ having suggest s.

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

The suggest with the highest $f(s, z_n)$ value among $S(d)$ is then denoted as $s(d)$. Word embedding is then applied to learn the feature vector of $s(d)$ representing d.

*1 In this paper, as the set V of vocabulary, we use the set of entry titles of the Japanese version of Wikipedia, where the version we used in this evaluation was downloaded in March 2014 and has about 1,407,000 entries.

*2 `http://gibbslda.sourceforge.net/`

Table 1: Query Focuses, # of Suggests, and # of Documents of the Dataset

query focus	# suggests	# valid suggests (Wikipedia only)	# valid suggests (Wikipedia +Web pages)	# total doc (training data)	# total doc (5 topics per query focus)	# valid doc (Wikipedia only)	# valid doc (Wikipedia +Web pages)
就活 (syukatsu)	925	575	671	13,222 (30.8 MB)	150	105	121
結婚 (kekkon)	947	637	694	14,413 (31.0 MB)	150	92	116

5.1.2 Word2Vec

The first approach to learn vector representation $v(d)$ of documents attempts to train a Word2Vec model so that the most frequency suggest $s(d)$ can be embedded and used 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, neural probabilistic language models are trained to maximize the likelihood $p(w_t, c)$ 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.

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

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

$$p(w_t|c) \equiv \frac{\exp(v_c \cdot \tilde{v}_{w_t})}{\sum_{w' \in V} \exp(v_c \cdot \tilde{v}_{w'})}$$

Computing the log likelihood becomes unacceptably expensive given a 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 loss below.

$$L^{SG} = \sum_{t=1}^{T} \sum_{c \in C_{w_t}} (\log \sigma(v_c \cdot \tilde{v}_{w_t}) + \sum_{k=1}^{K} \log \sigma(-v_c \cdot \tilde{v}_{\breve{w}'}))$$

Given some context h from the training data, log likelihood of every word w_t is optimized against its contrastive words $\breve{w}'$. 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. Similarity between documents are calculated as the cosine similarity of suggest word vectors $v(s(d))$.

$$sim(v(d), v(d')) = \frac{v(s(d)) \cdot v(s(d'))}{\| v(s(d)) \| \| v(s(d')) \|}$$

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.

5.1.3 Japanese Version of Wikipedia as the Training Data

As the skip-gram model is highly scalable to large training data we started with entry text from Wikipedia containing all available articles in Wikipedia.[*3] We extract the embedding vector $v(s(d))$ of suggest $s(d)$ as representation $v(d)$ of document d. After training is completed, we look up the trained model to find the vector $v(s(d))$ for every $s(d)$. One problem for such annotation is that not every suggest will be available for vector representation since some of the words are never present in the throughout Wikipedia texts. This leads to consequence that some documents fail to get representation $v(d)$ due to lack of embedding of $s(d)$. Therefore our algorithm ignores documents without valid $v(d)$ at run time and they will be taken special care of at evaluation time. Word2Vec trained with minimum count 5 embeds $s(d)$ for 105 and 92 documents with respect to individual query focus as listed in Table 1.

5.1.4 Japanese Version of Wikipedia along with Web Pages as the Training Data

As it now becomes clear that a well trained Word2Vec model[*4] lacks many suggests crucial to our task of learning vector representation of documents, we launched a second attempt to mitigate this problem by integrating the Wikipedia text and collected Web pages as training data. There are 14,413 and 13,222 appended Web pages in each query focus with roughly 30MB in size as indicated in Table 1. By appending the query focus text to the original Wikipedia data, we ensure that the new training data contains a lot more suggest words in its vocabulary because these documents are Web pages collected through annotated suggests as is mentioned in Section 2.2. As expected many more suggests are successfully recovered as Table 1 shows that 121 and 116 documents are assigned with valid $v(s(d))$ with the same minimum count of 5. However, it is not yet able to embed every suggest because a Web page d does not necessarily contain $s(d)$ explicitly in its context.

5.2 Doc2Vec based Measure

In addition to Word2Vec based models which learns vector representation of documents only based on the annotated suggest, we develop one more approach that directly learns documents as paragraph vectors considering

*3 This paper uses the Japanese version of Wikipedia updated by February, 2016 containing about 1 million entry pages with a total size of roughly 2.8 GB.

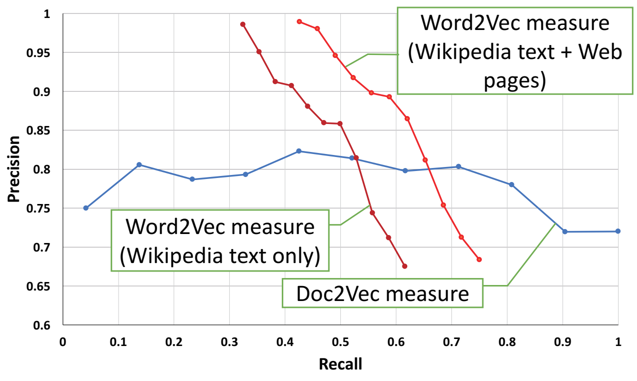
*4 `https://radimrehurek.com/gensim/models/word2vec.html` no suggest. Doc2Vec[*5] model [Le14] is a generalization of Word2Vec model which performs identical training mechanism to Word2Vec. Doc2Vec not only learns vectors for single words but also models vectors $p(d)$ for individual paragraphs in the 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 context, so that the paragraph ID is equivalent to another word from its softmax classifier perspective. We regulate every d to be a single paragraph so that $v(d) = p(d)$ and this model learns a valid $v(d)$ for every document in the query focus. For Doc2Vec based model we used the Web pages alone as the training data where 14,413 and 13,222 pages were used in "kekkon" and "syukatsu", as displayed in Table 1. Similarity between documents using Doc2Vec are calculated as the cosine similarity of $p(d)$.

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

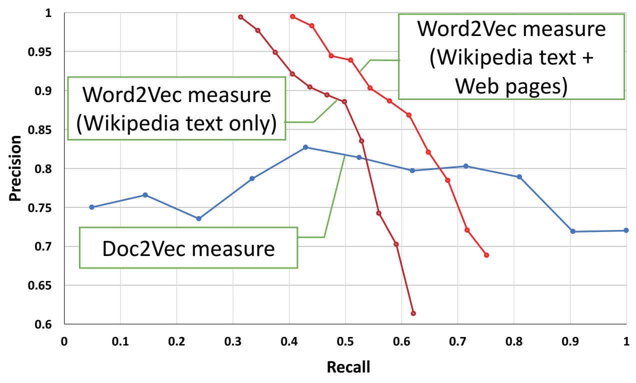
6. Evaluation

In this paper, we picked up 5 topic clusters from both query focuses, and for every topic, we selected top 30 documents based on their topic probability ranking $p(z_n \mid d)$. Our experimental dataset consists of altogether 300 documents from 2 query focuses as listed in Table 1, along with the total number of suggests per query focus. The Word2Vec based measures do not guarantee valid representation for every document. Our selection algorithm in Section 4. ignores such documents for candidate pairs. For example, only 694 out of 947 suggests were successfully embedded with "Wikipedia + Web pages" training data of "kekkon" and even fewer in "Wikipedia only" training data. This leads to 34 and 58 documents without valid $v(s(d))$ respectively. Therefore similarity comparison of $v(s(d))$ is not available for these documents. Still, all the documents are evaluated and documents $d \notin major\ doc(z_n, \theta_{lbd})$ are considered minor documents, meaning that documents without valid $v(d)$ are never selected as major ones by the algorithm. Evaluation of unsupervised subtopic labeling is given in terms of precision/recall calculated by comparing the algorithm prediction of the set $major\ doc(z_n, \theta_{lbd})$ to the set of major documents of manually created ground truth labels $labeled\ doc(z_n)$ for topic z_n. Moreover, precision and recall across topic clusters are calculated in macro average and micro average. The macro average is the mean of precision/recall of all topics while the micro average takes the sum of major document numbers from all topics. Precisions and recalls are evaluated for different possible θ_{lbd} values on the interval $[-1, 1]$ which is domain for cosine similarity, with increment of 0.02. Figure 1 displays this evaluation. The figure shows that combining Wikipedia text and query focus Web pages contributed to the overall performance as it discovers more available suggest embedding.

*5 `https://radimrehurek.com/gensim/models/doc2vec.html`



(a) Macro Average



(b) Micro Average

Figure 1: Evaluation Results (All Query Focuses and Topics)

7. Conclusion

The primary task of this paper is to propose the algorithm of selecting documents on major subtopics from existing topics given by the topic model. It presents three unsupervised approaches of learning distributed representation of documents. Two of them embed search engine suggests with Word2Vec model with different training data and the other is Doc2Vec model. Our algorithm exhibits different precision/recall characteristics in three representations. Evaluation indicates that suggest based Word2Vec measures tend to achieve higher precision yet with numerical instability as recall increases, due to imperfection of suggest allocation to all documents. Doc2Vec based measure ensures every document is well embedded and has more stable precision performance as recall goes higher.

References

[Blei03] Blei, D. M., Ng, A. Y. and Jordan, M. I.: Latent Dirichlet Allocation, *Journal of Machine Learning Research*, Vol. 3, pp. 993–1022 (2003).

[Le14] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proc. 31st ICML*, pp. 1188–1196 (2014).

[Mikolov13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Proc. NIPS*, pp. 3111–3119 (2013).