Constructing a Class-Based Lexical Dictionary using Interactive Topic Models

Kugatsu Sadamitsu, Kuniko Saito, Kenji Imamura and Yoshihiro Matsuo

NTT Cyber Space Laboratories, NTT Corporation
1-1 Hikarinooka, Yokosuka-shi, Kanagawa, 239-0847, Japan
{sadamitsu.kugatsu, saito.kuniko, imamura.kenji, matsuo.yoshihiro}@lab.ntt.co.jp

Abstract

This paper proposes a new method of constructing arbitrary class-based related word dictionaries on interactive topic models; we assume that each class is described by a topic. We propose a new semi-supervised method that uses the simplest topic model yielded by the standard EM algorithm; model calculation is very rapid. Furthermore our approach allows a dictionary to be modified interactively and the final dictionary has a hierarchical structure.

This paper makes three contributions. First, it proposes a word-based semi-supervised topic model. Second, we apply the semi-supervised topic model to interactive learning; this approach is called the Interactive Topic Model. Third, we propose a score function; it extracts the related words that occupy the middle layer of the hierarchical structure. Experiments show that our method can appropriately retrieve the words belonging to an arbitrary class.

Keywords: Interactive Topic Models, Interactive Unigram Mixtures, Lexical dictionary

1. Introduction

Many NLP applications, such as recommendation engines, demand a class-based related word dictionary. The related words are broadly similar word set about one class, e.g. for the class ”*Car*”, the related word set includes {*Toyota, engine, hybrid*} et cetera. However, class definitions are often changed in response to application or user needs. One application assigns word *SUV* and word *motorcycle* to the same class (related), while another differentiates them (not related). Although the hierarchical or graph based structure (e.g. WordNet) or multi-labeling (e.g. Wikipedia) can provide a rough solution, there is no universal definition of hierarchical structure or multi-labeling that can satisfy every application.

This paper describes a way of constructing arbitrary class-based word dictionaries on semi-supervised topic models trained against non-annotated text corpus; we assume that each class is described by a topic. Because unsupervised topic models lack control by human intent, we adapt semi-supervised learning using a few supervised words and interactive learning.

This paper makes three contributions. First, it proposes the word-based semi-supervised topic model. In a previous study about semi-supervised methods (Nigam et al., 2000), document labels are given as supervised data. Because our purpose is constructing a dictionary, it is more naturally that we use words as clues for classes. Our semi-supervised method uses the simplest topic model yielded by the standard EM algorithm, so model calculation is very rapid.

Second, we apply the semi-supervised topic model to interactive learning. After learning topic models that include the defined class “*Finance*”, the user might think that “*Finance*” topic should be split into “*Bank*” and “*Insurance*”. Our interactive methods allow such user’s interaction. Furthermore, our models not only modify the output of topic models, but also construct a hierarchical structure.

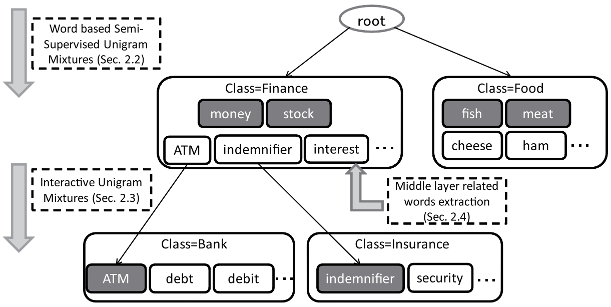

Figure 1: The abstract illustration of hierarchical topic structure and correspondence sections. The words in colored boxes are supervised words and the words in white boxes are extracted words by our method.

Third, we propose a score function; it extracts the related words that occupy the bottom or middle layer of the hierarchical structure. Experiments show that our method can appropriately retrieve the words belonging to an arbitrary class. Figure 1 illustrates the abstraction of our topic models and the relevant sections.

2. Construct arbitrary class-based word dictionaries using Interactive Topic Models

For the purpose of extracting related words, we can use topic models whose topic is assumed to correspond to a class. Previous works describe several experiments on extracting characteristic words for topic model analysis (Nigam et al., 2000; Hofmann, 1999; Blei et al., 2003). They use the modified parameters of topic models that describe the relation between each word and each topic. Basically, our score function for extraction related words is also based on the parameters of topic models.

In this section, we describe how to extract related words from topic models and then propose semi-supervised topic models and new interactive topic models.

2.1. Semi-supervised EM algorithms

In this section, we review the simplest semi-supervised topic model called Unigram Mixtures (Nigam et al., 2000). Unigram Mixtures are defined as

$$p(D) = \prod_{d=1}^{D} \sum_{z} p(z) \prod_{v} p(v|z)^{n(v,d)}, \qquad (1)$$

where D is set of documents, d indicates a document, z is a hidden topic of a document, v is a word type, $n(v,d)$ is the word count of v in document d. $p(z)$ and $p(v|z)$ are the model parameters. Their approach is to use the standard EM algorithm to estimate Unigram Mixtures. The estimation is achieved by computing the following formulae,

$$\begin{aligned} p(v|z) &= \frac{\sum_d n(v,d)p(z|d)}{\sum_v \sum_d n(v,d)p(z|d)} & (2) \\ p(z) &= \frac{\sum_d p(z|d)}{|D|}, & (3) \end{aligned}$$

where $p(z|d)$ is called a posterior probability of topic z about document d. A posterior probability $p(z|d)$ is calculated in E-step by the following formula,

$$p(z|d) = \frac{p(z)\prod_v p(v|z)^{n(v,d)}}{\sum_z p(z)\prod_v p(v|z)^{n(v,d)}}. \qquad (4)$$

Nigam also proposed Laplace smoothing as follows,

$$\begin{aligned} p(v|z) &= \frac{1 + \sum_d n(v,d)p(z|d)}{|V| + \sum_v \sum_d n(v,d)p(z|d)} & (5) \\ p(z) &= \frac{1 + \sum_d p(z|d)}{|Z| + |D|}. & (6) \end{aligned}$$

This added parameter, “1”, is called pseudo count. It is equal to adding a document that has all words and its one of posterior probability $p(z|d)$ is 1. Applying Nigam's ideas straight forwardly to our supervised format, we make pseudo documents. A pseudo document consists of only supervised words belong to one class. For example, we assume that the supervised words of the domain about “*Finance*” are {“*bank*”, “*interest*”}, and are regarded as one pseudo document including words themselves with BOW expression like $d_{s1} = (\text{“}\textit{bank}'' : 1), d_{s2} = (\text{“}\textit{interest}'' : 1)$, where d_s indicates a document including some supervised words. The update parameters assumed with pseudo documents are calculated as follows,

$$p(v|z) = \frac{N_v + \sum_d n(v,d)p(z|d)}{\sum_{v'}\{N_{v'} + \sum_d n(v',d)p(z|d)\}}, \qquad (7)$$

where N_v is the number of pseudo documents that include word v and can adjust the weight of learning for supervised words.

Unfortunately, this method is not appropriate for constructing semi-supervised topic models because the effect of each document on EM estimation is not so large. Although the number of pseudo documents N_v can be adjusted arbitrarily, we don't know which number is best for each word v.

Nigam et al. also extended Unigram Mixtures to realize document classification in a semi-supervised manner. In the classification task, we always input some supervised data including the data itself and annotated labels. After that, the supervised classifier is trained by the data and labels. Semi-supervised methods handle not only supervised documents but also unsupervised documents, i.e. those without annotation labels. Nigam's goal is document level classification and the supervised resources consist of documents. Supervised data is treated as document data whose posterior probability is deterministic decided as either 1.0 or 0.0 by using supervised document label and remaining data is estimated by conventional EM algorithms. Unlike his work, our goal is to construct a word dictionary and the supervised resources consist of words. For example, we input “*Finance*={*bank, interest*}, *Food*={*sugar, ham*} ” as supervised class definitions, i.e. supervised words for each class. Because Nigam's semi-supervised methods are based on supervised data with document labels, they can not be applied to our task.

2.2. Word based Semi-Supervised Unigram Mixtures

In this section, we propose a way of constructing semi-supervised topic models from a small set of supervised words. To achieve this, we introduce supervised posterior probability ($p_s(z|d_s)$) of topic z according to document d_s including supervised words. The supervised posterior probability is calculated as

$$p_s(z|d_s) = \frac{n_{d_s}(z)}{N_{d_s}}, \qquad (8)$$

where $n_{d_s}(z)$ is the number of supervised words in document d_s that belong to topic z. N_{d_s} is the number of supervised words belonging to any topic, $N_{d_s} = \sum_z n_{d_s}(z)$. For example, we consider two documents, $\{bank, interest\} \in d_{s1}$ and $\{bank, sugar\} \in d_{s2}$. The supervised posterior probability of d_{s1} and d_{s2} is calculated as $p_s(z = \text{''}\textit{Finance}\text{''}|d_{s1}) = 1$ and $p_s(z = \text{''}\textit{Finance}\text{''}|d_{s2}) = 0.5$, $p_s(z = \text{''}\textit{Food}\text{''}|d_{s2}) = 0.5$ [1].

Furthermore, the supervised posterior probabilities, $p_s(z|d_s)$, are interpolated from the calculated posterior probabilities, interpolated posterior probability $p_i(z|d_s)$ is calculated as

$$p_i(z|d_s) = w \cdot p_s(z|d_s) + (1-w) \cdot p_c(z|d_s). \qquad (9)$$

In contrast to the supervised posterior probability, this interpolation method reduces the risk that the modeling will yield mismatching topic models. In particular, if ambiguous supervised words are given, whether consciously or not, they strengthen the attachment of wrong labels to documents including such ambiguous supervised words. These initial mistakes yield erroneous topic models. Since the interpolation method balances the supervised posterior probability against calculated posterior probability, it reduce this risk.

[1] z is a random variable whose sample space is represented as a discrete variable, not explicit words.

In the initial EM iteration, we set 1 to interpolation weight w which means we use only the supervised posterior probability. Interpolation weight w is decreased with each iteration. In early iterations, w takes a high value to permit to model learning to closely approach the supervised structure. In later iterations, w is given a low value to adjust the total balance of model parameters from the perspective of probabilistic adequacy.

After processing E-step, M-step is performed to update parameter $p(v|z)$ in the same way as conventional EM algorithm applications, i.e. Eq. (2). We refer to this method as "SSUM".

After learning SSUM, we can rank and extract the top N words as the related words for each topic by using score function

$$score(v, z) = \frac{p(v|z)}{p_{uni}(v)}, \tag{10}$$

where $p_{uni}(v)$ is the global unigram probability of all words in all documents.

2.3. Interactive Topic Models as Extended Semi-Supervised Unigram Mixtures

After creating a topic model, we may find that some topics are not intuitive. For example, the topic model includes the topic "*Finance*" and we want to separate it into "*Bank*" and "*Insurance*". This is possible by inputting additional clues; e.g. topic "*Bank*" (*ATM, deposit*) and "*Insurance*" (*hospitalization, 401k*) . Our purpose is to permit free control of a topic model through human or machine interaction. For an interactive method, the factor of calculation overhead is very important because a user has to wait for system feedback before interaction is possible. From these perspectives, we propose a new interactive topic model.

Topic model modification is now possible with the recent proposal of the Interactive Topic model (ITM) (Hu and Boyd-graber, 2011). ITM is based on the Dirichlet Forest prior models (Andrzejewski et al., 2009) and is estimated by Gibbs sampling. ITM makes it possible to accept the alterations input by users and revise the topic model accordingly. In the previous example, ITM modifies the hidden topic assignment of supervised words (*ATM* or *hospitalization*) in Gibbs sampling results and updates the models using statistics updated by the new assignment. Note that the main purpose of ITM is modifying the topic models. On the other hand, our main purpose in this section is segmentation for coarse grain topics. Although ITM can modify a topic model, the calculation cost is high because it uses Gibbs sampling. If user-interactivity is to be well accepted, we need to raise the response speed.

To develop faster interactive models, we apply SSUM. After creating a topic model using SSUM, documents are clustered in topics according to memorized posterior probability $p(z|d)$. A document is assigned (clustered) to topic z when its posterior probability about topic z is $p(z|d) \geq 0.5$. The likelihood is defined as follows.

$$p(D_z) = \prod_{d \in D_z} \sum_{z' \in S_z} p(z') \prod_{v} p(v|z')^{n(v,d)}, \tag{11}$$

where D_z is the subset of documents whose posterior probability $p(z|d) \geq 0.5$, S_z is a set of sub-topics of parent topic z and z' is a sub-topic of parent topic z. Interactive updating involves using the new clues (words) to re-estimate the clusters as in ITM; for example, the supervised words "*ATM,deposit*" for class "*Bank*". This interactive method is faster than Gibbs sampling because only the standard EM algorithm is used.

When we use SSUM to support interactivity, the initial parameters are very important for modeling sub-topics appropriately. If the initial parameters are given at random, the model might converge on an inadequate local minima. EM algorithms separate each topic distribution (multinomial, in this case) as far as possible and lack a general topic. For example, after we get the first topic models using SSUM, we apply SSUM again to the "*Finance*" topic to model the child topics with initial parameters at random. The output child topics are "*Sports*" and "*Food*" topics even though the parent topic was "*Finance*" and the supervised words were appropriate. To avoid converging on inadequate local minima, we set the initial parameters to the parent topic model parameters (For child topics "*Bank*" and "*Insurance*", the parent topic is "*Finance*"). These initial parameters are used as bias to prevent the parameters of the child topic models from converging on inadequate local minima. We refer to this model as "IUM".

The related words positioned in the child topic are ranked and extracted by the following score function

$$score_c(v, z') = \frac{p(v|z')}{p_{uni}(v)}, \tag{12}$$

If we treat $p_{uni}(v)$ as a unigram model of the parent subset about documents, the related words tend to be selected as far as parent topic. For example, for the parent topic of "*Finance*", the child topics can model "*Sports*" or "*Politics*" which are far from the parent topic "*Finance*". Using global unigram models, $p_{uni}(v)$, the words similar to the parent topic are given higher score.

2.4. Extracting Related Words from the Middle Layer Topics

In the hierarchical topic models, the related word in the bottom layer can be extracted with the scoring function discussed in the previous section. We propose here a score function that extracts the related words positioned in the middle layer of the hierarchical structure. This extraction uses different criteria from that used in the bottom layer. For example, consider parent topic "*Finance*" and child topics "*Bank*" and "*Insurance*". The word "*ATM*" is an adequate related word for bottom layer topic "*Bank*" and the word "*indemnifier*" is adequate related word for bottom layer topic "*Insurance*". However, both words are too specific to be adequate for the middle layer topic "*Finance*". For middle layer topic "*Finance*", "*deposit*" or "*interest*" are considered to be more suitable words.

For extracting middle layer related words, we use the variance of words in different layers. The middle layer related words have to represent the common features of child topics, but not specific child topics. Furthermore, the middle layer related words have to represent different features of the parent topic. To express these properties in the score

Table 1: Results: precision of "*Food*" class.

class	Feedstuff		Sugar		Milling		Food oil		Alcohol		Bread/Snack		Ham		Flavoring		Dairy		Others	
#seed	4		7		6		3		4		17		8		17		4		30	
method	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2
@10	1.0	1.0	0.5	0.7	0.9	0.8	0.3	0.2	1.0	1.0	1.0	1.0	0.4	0.9	0.3	0.8	0.2	0.9	0.3	1.0
@50	0.9	1.0	-	0.64	0.88	0.88	-	-	0.9	1.0	1.0	0.9	-	0.68	-	0.86	-	0.9	-	1.0
@100	0.7	0.95	-	0.52	0.54	0.54	-	-	0.7	0.95	0.75	0.9	-	-	-	0.73	-	0.5	-	0.85

Table 2: Results: precision of "*Finance*" class.

class	City Bank		Local Bank		Trust Bank		Brokerage		Insurance		Lease		Leased immovables		Sold immovables	
#seed	4		95		3		16		3		29		13		24	
method	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2
@10	0.4	1.0	0.5	1.0	0.6	1.0	0.8	0.9	0.9	0.7	0.0	0.0	1.0	0.3	1.0	0.4
@50	-	1.0	-	0.7	0.72	0.8	0.56	0.88	0.98	0.84	-	-	0.4	-	0.8	-
@100	-	0.95	-	0.75	0.61	0.55	0.58	0.69	0.79	0.62	-	-	-	-	0.6	-

Table 3: Results: precision of middle layer classes.

class	*Food*		*Finance*	
#seed	100		187	
method	BL	Var.	BL	Var.
@10	0.0	0.7	0.0	0.5
@50	-	0.74	-	0.4
@100	-	0.62	-	0.0

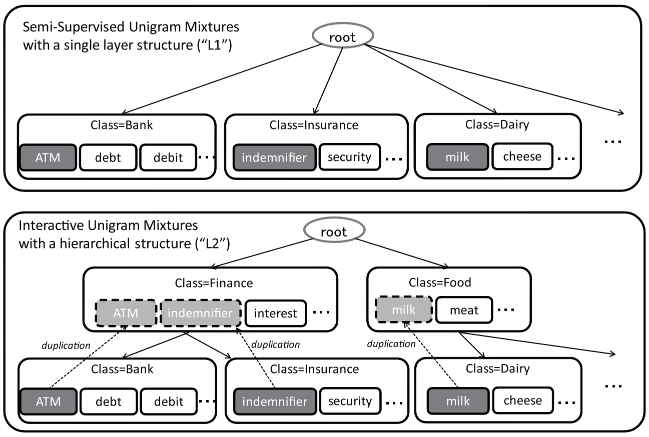

Figure 2: The experimental settings of semi-supervised Unigram Mixtures with a single layer structure (upper side) and Interactive Unigram Mixtures with a hierarchical structure (lower side).

function, at first, we define the parent variance and child variance as

$$\sigma_p(v) = (p(v|z) - p(v|z_p))^2 \quad (13)$$

$$\sigma_c(v) = \sum_{z_c}(p(v|z_c) - p(v|z))^2, \quad (14)$$

where z is the focusing middle layer topic and z_p is the parent topic of topic z and z_c is the child topic of z. The parent variance represents the characteristics of difference z from the parent topic z_p while the child variance represents the common characteristics of topic z and its child topics z_c.

Finally, we use the score function,

$$score_m(v, z) = \frac{\sigma_p}{\sigma_c}, \quad (15)$$

for ranking and extracting related words for middle layer topics. This function give high score to the word whose parameter is far from parent topic parameter (large variance) and which appear equally in the child topics (small variance).

3. Experiments and Results

In this section, we examine the dictionaries constructed by the proposed topic model with pre-defined structure. For confirming the success of the proposed interactive update process, we use supervised data whose hierarchical structure consists of 1st and 2nd layer classes. We compare SSUM with a single layer structure (L1) and IUM with a hierarchical layer structure (L2). This experimental settings are listed in Figure 2.

The supervised resource is our in-house thesaurus about Japanese companies; it has a hierarchical style with 1st and 2nd layer classes. The 1st layer of the thesaurus has 22 classes and the 2nd layer has 135 classes extended from 17 classes in the 1st layer. A few supervised words are given for all topics belonging to 2nd layer topics. The numbers of supervised words are partially listed in the 2nd row of Table 1 and Table 2 (#seed) as examples of 2nd layer classes included in 1st layer classes; "*Food*" and "*Finance*". The supervised words in the first layer are duplicated and merged from second layer topics. For SSUM with a single layer, the 1st layer from the thesaurus is ignored for the IUM situation and only 2nd layer topics are modeled using 2nd layer supervised words. In the case of using IUM, the 1st layer is modeled with SSUM and the 2nd layer is modeled with IUM. This process is reasonable as the interactive update of the 2nd layer given that the 1st layer topics have excessively coarse grain and the intent to separate the 2nd layer classes.

We set the mixture number of topic model as follows. If the mixture number exactly equals the number of classes we want, the other topics not belonging to any supervised topics are admixed with supervised topics and the constructing confusable topics. Therefore, the baseline methods simultaneously model not only 140 classes ($(22 - 17) + 135$) but also the other topics; total topic number is 200. The number 200 was set in preliminary experiments. For IUM, we first model the 1st layer thesaurus by 100 mixtures with SSUM including 22 supervised topics. After that, each semi-supervised topic is modeled again for the 2nd layer of thesaurus by using the proposed methods and the exact number of mixtures with IUM (Sec. 2.3.).

Finally, we extracted related words given top 100 scores as yielded by the proposed score function (Eq. (10) and Eq. (12)).

We used 9 years worth of the Japan Economic Newspaper printed from 1990 to 1995, 2000, 2001 and 2002 as the corpus ($1,593,950$ documents). All documents were tokenized by JTAG (Fuchi and Takagi, 1998), chunked for named entities by CRFs using Minimum Classification Error rate (Suzuki et al., 2006). We used only those words that occurred over 20 times in the corpus. Total number of words was $425,517,354$ and total number of word types was $126,218$.

Two evaluators judged each related word as to whether it was relevant to the target class or not. The words occupying ranks 1 to 10 were evaluated against all words, those in ranks 11 to 100 were evaluated against each 5th word. The evaluation was stopped when the accuracy was lower than 50% at rank 10 and rank 50. The results are shown in Table 1 and Table 2; they show 2 classes from the 1st layer, "*Food*" and "*Finance*".

For the "*Food*" class, IUM had better accuracy than SSUM for most 2nd layer classes except for "*Food oil*". For the class of "*Ham*", SSUM extracted many words about "*Sports*" because the company "*Nippon Ham*" is also the name of a professional Japanese baseball team. IUM avoided such mistakes because the 1st layer topic "*Food*" was far from "*Sports*".

For the "*Finance*" class, these two methods were more competitive. IUM completely dominated SSUM for "*City Bank*", "*Local Bank*" and "*Brokerage*" classes. The precisions of two classes for "*Immovables*" were lower. This is because these two classes were actually far from the "*Finance*" class, and IUM failed to capture these two classes when constructing the 1st layer. This is considered to be the reverse case of "*Ham*" class. To resolve this problem, our model should indicate the distance between topics and adequacy of the defined structure to users and revise the topic models accordingly.

There is a general problem with regard to threshold. The results of each class have a large variance. In the "*Feedstuff*" class, the accuracy of the top 100 words does not drop below 0.95. On the other hand, in the "*Ham*" class, the accuracy drops under 0.68 for the top 50 words. Finding the appropriate threshold is important remaining problem.

Finally, we compare the middle layer related words shown in Table 3. In both classes, the proposed variance-based methods ("Var.", Eq. (15)) yielded more appropriate extraction than the baseline score ("BL", as for the bottom layer score function, Eq. (10)). "BL" score function gives high value to the words belonging to a specific child topic. For example, "Japanese rice chips industry association" (*Zenkoku-Beika-Kogyo-Kumiai*) is given high value in "*Food*" topic, although, it is related more strongly to the child topic "*Bread/Snack*". On the other hand, "Var" score function extracts "*fluid*" (*sui bun*) or "*low temperature*" (*tei on*) which are common related words of "*Food*" topic.

4. Related works

Some previous works proposed the use of wikipedia categories to extract a universal ontology (Ponzetto and Strube, 2007; Nagata et al.,). They focused on the structure of ontology, for example hypernyms and hyponyms. Because our purpose is to semi-automatically construct ontologies that depend on each application, we prefer that the ontology have a controllable structure.

Nagata et al. (Nagata et al.,) proposed a method for matching categories in Japanese Wikipedia to categories in Japanese Goi-Taikei (Ikehara et al., 1997). Ponzetto and Strube (2007) proposed a method to extract semantic relations between wikipedia categories such as *is-a* relations. In the relation extraction area, YAGO targeted the construction of an ontology with arbitrary type definition using wikipedia and WordNet (Suchanek et al., 2007). All of them target the construction of a universal ontology or structure; we, on the other hand, focus on constructing application-specific ontologies.

Snow et al.(2006) proposed a method for adding entries to the synset of WordNet. The structure of the ontology is given that of WordNet itself. Although their approach is applicable for large structures with many entries, it is not suitable for small numbers of supervised entries. Our proposal can extract new entries even when the size of supervised entries is very small.

5. Conclusion

We proposed word-based semi-supervised Unigram Mixtures and Interactive Unigram Mixtures for constructing arbitrary lexical dictionaries. First, we proposed a word-based semi-supervised topic model, semi-supervised Unigram Mixtures. Second, we proposed a fast interactive topic model, Interactive Unigram Mixtures which applies semi-supervised Unigram Mixtures to interactive learning. Third, we proposed a score function; it extracts the related words that occupy the bottom or middle layer of the hierarchical structure. Our approach was shown to be more accurate in extracting related words belonging to arbitrary classes.

The extraction of related words faces a kind of threshold problem. This is because each component of the topic model expresses a different topic granularity. Finding the appropriate threshold is an important remaining problem and using the score function as the threshold is an attractive possibility.

Furthermore, we will try to construct different structure dictionaries using our methods and confirm their effectiveness.

6. References

David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating Domain Knowledge into Topic Modeling via Dirichlet Forest Priors. In *Proceedings of the ICML.*, volume 382, pages 25–32.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Takeshi Fuchi and Shinichiro Takagi. 1998. Japanese Morphological Analyzer using Word Co-occurrence-JTAG.

In *Proceedings of the COLING-ACL Conference*, pages 409–413.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57.

Yuening Hu and Jordan Boyd-graber. 2011. Interactive Topic Modeling. In *Proceedings of the 49th ACL-HLT*, pages 248–257.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi, editors. 1997. *Nihongo Goi-Taikei - a Japanese Lexicon (in Japanese)*. Iwanami Shoten.

Masaaki Nagata, Yumi Shibaki, and Kazuhide Yamamoto. 2010. Using goi-taikei as an upper ontology to build a large-scale japanese ontology from wikipedia. In *Proceedings of the 6th Workshop on Ontologies and Lexical Resources, COLING* , pages 11–18.

Kamal Nigam, Andrew K Mccallum, Sebastian Thrun, and Tom Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2):103–134.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22nd Conference on Advancement of Artificial intelligence (AAAI)*, pages 1440–1445.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of the 16th International World Wide Web (WWW) Conference*, pages 697–706.

Jun Suzuki, Erik McDermott, and Hideki Isozaki. 2006. Training Conditional Random Fields with Multivariate Evaluation Measures. In *Proceedings of the COLING-ACL Conference*, pages 217–224.