Formalizing Word Sampling for Vocabulary Prediction as Graph-based Active Learning

Yo Ehara*
National Institute of Information and Communications Technology
ehara@nict.go.jp

Yusuke Miyao
National Institute of Informatics
yusuke@nii.ac.jp

Hidekazu Oiwa
Issei Sato
Hiroshi Nakagawa
The University of Tokyo
{oiwa,sato}@r.dl.itc.u-tokyo.ac.jp
nakagawa@dl.itc.u-tokyo.ac.jp

Abstract

Predicting vocabulary of second language learners is essential to support their language learning; however, because of the large size of language vocabularies, we cannot collect information on the entire vocabulary. For practical measurements, we need to sample a small portion of words from the entire vocabulary and predict the rest of the words. In this study, we propose a novel framework for this sampling method. Current methods rely on simple heuristic techniques involving inflexible manual tuning by educational experts. We formalize these heuristic techniques as a graph-based non-interactive active learning method as applied to a special graph. We show that by extending the graph, we can support additional functionality such as incorporating domain specificity and sampling from multiple corpora. In our experiments, we show that our extended methods outperform other methods in terms of vocabulary prediction accuracy when the number of samples is small.

1 Introduction

Predicting the vocabulary of second language learners is essential to support them when they are reading. Educational experts have been continuously studying methods for measuring the size of a learner's vocabulary, i.e., the number of words the learner knows, over the decades (Meara and Buxton, 1987; Laufer and Nation, 1999). Ehara et al. (2012) formalized a more fine-grained measurement task called *vocabulary prediction*. The goal of this task is to predict whether a learner knows a given word based on only a relatively small portion of his/her vocabulary. This vocabulary prediction task can be further used for predicting the readability of texts. By predicting vocabulary unknown to readers and showing the meaning of those specific words to readers, Ehara et al. (2013) showed that the number of documents that learners can read increases.

Word sampling is essential for vocabulary prediction. Because of the large size of language vocabularies, we usually cannot collect information on the entire vocabulary. For practical measurements, we inevitably need to sample a small portion of words from the entire vocabulary and then predict the rest. We refer to this sampling technique as word sampling.

Word sampling can greatly affect the performance of vocabulary prediction. For example, if we consider only short everyday general domain words such as "cat" and "dog" as samples, the rest of the vocabulary is difficult to predict since learners likely know most of these words. To more accurately measure a learner's vocabulary, we ideally must sample words that are representative of the entire set of words. More specifically, we wish to sample words such that if a learner knows these words, he/she is likely to know the rest of the words in the given vocabulary, and vice versa.

To our knowledge, however, all current studies have relied on a simple heuristic method. In this heuristic method, educational experts first somehow create groups of words with the aim that the words in a group are of similar difficulty for learn-

*The main body of this work was done when the first author was a Ph.D. candidate in the University of Tokyo and the paper was later greatly revised when the first author was a JSPS (Japan Society for the Promotion of Science) research fellow (PD) at National Institute of Informatics. See http://yoehara.com/ for details.

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1374–1384,
October 25-29, 2014, Doha, Qatar.

ers. To create groups of words, the experts typically make use of word frequencies and sometimes manually reclassify words based on experience. Next, a fixed number of words are randomly sampled from each group via a uniform distribution. We call this approach *heuristic word sampling*.

In this study, we propose a novel framework that formalizes word sampling as *non-interactive* graph-based active learning based on weighted graphs. In our approach, nodes of a graph correspond to words, whereas the edge weights show how similar the difficulty levels of a word pair are. Unlike *interactive* active learning algorithms used in the NLP community, which use expert annotators' human labels for sampling nodes, non-interactive active learning algorithms exclude expert annotators' human labels from the protocol (Ji and Han, 2012; Gu and Han, 2012). Given a weighted graph and using only its structure, without human labels, these algorithms sample nodes that are important for classification with algorithms called *label propagation*. Excluding annotators' human labels from the protocol is beneficial for educational purposes since learners can share the same set of sampled words via, for example, printed handouts.

Formalizing the current methods as non-interactive graph-based active learning enables us to extend the sampling methods with additional functionality that current methods cannot handle without applying burdensome manual heuristics because we can flexibly design the weighted graphs fed to the active learning algorithms. In our framework, this extension is achieved by extending the graph, namely, our framework can handle *domain specificity* and *multiple corpora*.

Domains are important when one wants to measure the vocabulary of learners. For example, consider measuring non-native English speakers taking computer science graduate courses. We may want to measure their English vocabulary with an emphasis on computer science rather than their general English vocabulary. However, such an extension is impossible via current methods, and thus it is desirable to sample algorithms to be able to handle domain specificity. Our framework can incorporate domain specificity between words in the form of edges between such words.

Handling multiple corpora is important when we cannot single out which corpus we should rely on. The current technique used by educational experts to handle multiple corpora is to heuristically integrate multiple frequency lists from multiple corpora into a single list of words; however, such manual integration is burdensome. Thus, automatic integration is desirable. Our framework converts multiple corpora into graphs, merges these graphs together, and then samples from the merged graph.

Our contributions as presented in this paper are summarized as follows:

1. We formalize word sampling for vocabulary prediction as graph-based active learning.

2. Based on this formalization, we can perform more flexible word sampling that can handle domain specificity and multiple corpora.

The remaining parts of this paper are organized as follows. In §2, we explain the problem setting in detail. We first explain how existing heuristic word sampling works and how it relies on the *cluster assumption* from the viewpoint of graphs. Then, we introduce existing graph-based non-interactive active learning methods. In §3, we show that the existing heuristic word sampling is merely a special case of a non-interactive active learning method (Gu and Han, 2012). Precisely, the existing sampling is identical to the case where a special graph called a "multi-complete graph" is fed to a non-interactive active learning method. Since this method can take any weighted graphs other than this special graph, this immediately leads to a way of devising new sampling methods by modifying graphs. §4 explains exactly how we can modify graphs for improving active learning. §5 evaluates the proposed method both quantitatively and qualitatively, and §6 concludes our paper.

2 Problem Setting

2.1 Heuristic Word Sampling

A simple vocabulary estimation technique introduced by educational experts is to use the frequency rank of words in a corpus based on the assumption that learners using words with similar frequency ranks have a similar vocabulary (Laufer and Nation, 1999). In accordance with this assumption, they first group words by frequency ranks in a corpus and then assume that words in each group have a similar vocabulary status. For example, they sampled words as follows:

1. Rank words by frequency in a corpus.

2. Group words with frequency ranks from 1 to $1,000$ as Level 1000, words with frequency ranks from $1,001$ to $2,000$ as Level 2000, and so on.

3. Take 18 samples from Level 1000, another 18 samples from Level 2000, and so on.

The rationale behind this method is to treat high-ranked and low-ranked words separately rather than sample words from the entire vocabulary. After sampling words, this sampling method can be used for various measurements; for example, Laufer and Nation (1999) used this method to estimate the size of the learners' vocabulary by simply adding $1,000 * \frac{\text{Correctly answered words}}{18}$ for each level.

2.2 Cluster Assumption

In the previous subsection, we noted that existing word sampling methods rely on the assumption that *words with similar frequency ranks are known to learners whose familiar words are similar each other*. This assumption is known as the cluster assumption in the field of graph studies (Zhou et al., 2004).

To further describe the cluster assumption, we first define graphs. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes (vertices) $\mathcal{V}$ and a set of edges $\mathcal{E}$. Here, each node has a *label*, and each edge has a *weight*. A label denotes the category of its corresponding node. For example, in binary classification, a label is taken from $\{+1, -1\}$. A weight is a real value; when the weight of an edge is large, we describe the edge as being heavy.

The cluster assumption is an assumption that *heavily connected nodes in a graph should have similar labels*. In other words, the cluster assumption states that weights of edges and labels of nodes should be consistent.

We explain how the cluster assumption relates to our task. In our application, each node corresponds to a word. Labels of the nodes in a graph denote the vocabulary of a learner. If he/she knows a word, the label of the node corresponding to the word is $+1$; if not, the label is -1. The cluster assumption in our application is that the heavier the edge, the higher the similarity between users familiar with the two words.

In this manner, existing word sampling methods implicitly assume cluster assumption. This is therefore the underlying approach for reducing the word sampling problem into graph-based active learning. Since graphs allow for more flexible modeling by changing the weights of edges, we expect that more flexible word sampling will be enabled by graph-based active learning.

2.3 Label Propagation

Since the graph-based active learning algorithms are based on *label propagation* algorithms, we will explain them first. Basically, given a weighted graph, label propagation algorithms classify their nodes in a weakly supervised manner. While the graph-based active learning algorithm that we are trying to use (Gu and Han, 2012) does not use label propagation algorithms' outputs directly, it is tuned to be used with a state-of-the-art label propagation method called Learning with Local and Global Consistency (LLGC) (Zhou et al., 2004).

Label propagation algorithms predict the labels of nodes from a few manually supervised labels and graph weights. To this end, label propagation algorithms follow the following steps. First, humans label a small subset of the nodes in the graph. This subset of nodes is called the set of *labeled nodes*, and the remaining nodes are called *unlabeled nodes*. Second, label propagation algorithms propagate labels to the unlabeled nodes based on edge weights. The rationale behind label propagation algorithms lies in cluster assumption; as label propagation algorithms assume that two nodes connected by a heavily weighted edge should have similar labels, more heavily weighted edges should propagate more labels.

We formalize Learning with Local and Global Consistency (LLGC) (Zhou et al., 2004), one of the state-of-the-art label propagation methods. Here, for simplicity, suppose that we want to perform binary classification of nodes. Let N be the total number of nodes in a graph. Then, we denote labels of each node by $\mathbf{y} \stackrel{\text{def}}{=} (y_1, \ldots, y_N)^\top$. For unlabeled nodes, y_i is set to 0. For labeled nodes, y_i is set to $+1$ if the learner knows a word, -1 if not. We also introduce a label propagation (LP) *score vector* $\mathbf{f} = (f_1, \ldots, f_N)^\top$. This LP score vector is the output of label propagation and is real-valued. To obtain the classification result from this real-valued LP score vector for an unlabeled node (word) i, the learner is predicted to know the word i if $f_i > 0$, and he/she is predicted to be unfamiliar with the word if $f_i \leq 0$.

Next, we formally define a normalized graph-Laplacian matrix, which is used for penalization based on the cluster assumption. Let an $N \times N$ -sized square matrix $\mathbf{W}$ be a *weighted adjacency matrix* of $\mathcal{G}$. $\mathbf{W}$ is symmetric and non-negative definite; its diagonal elements $\mathbf{W}_{i,i} = 0$ and all other elements are non-negative[1]. The graph Laplacian of a normalized graph, known as a *normalized graph Laplacian* matrix, is defined as $\mathbf{L}^{\text{norm}}_{\mathbf{W}} \stackrel{\text{def}}{=} \mathbf{I} - \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}_{\mathbf{W}}^{-\frac{1}{2}}$. Here, $\mathbf{D}_{\mathbf{W}}$ is defined as a diagonal matrix whose diagonal element is $(\mathbf{D}_{\mathbf{W}})_{i,i} \stackrel{\text{def}}{=} \sum_{j=1}^{|\mathcal{V}|} \mathbf{W}_{i,j}$, and $\mathbf{I}$ denotes the identity matrix of the appropriate size. Note that a normalized graph Laplacian $\mathbf{L}^{\text{norm}}_{\mathbf{W}}$ depends on the weighted adjacency matrix $\mathbf{W}$.

Then, LLGC can be formalized as a simple optimization problem as shown in Equation 1.

$$\min_{\mathbf{f}} \|\mathbf{f} - \mathbf{y}\|_2^2 + \mu \mathbf{f}^\top \mathbf{L}^{\text{norm}}_{\mathbf{W}} \mathbf{f} \qquad (1)$$

Equation 1 consists of two terms. Intuitively, the first term tries to make the LP score vector, the final output $\mathbf{f}$, as close as possible to the given labels $\mathbf{y}$. The second term is designed to meet the cluster assumption: it penalizes the case where two nodes with heavy edges have very different LP scores. $\mu > 0$ is the only hyper-parameter of LLGC: it determines how strong the penalization based on the cluster assumption should be. Thus, in total, Equation 1 outputs an LP score vector $\mathbf{f}$ considering both the labeled input $\mathbf{y}$ and the cluster assumption of the given graph $\mathbf{W}$: the heavier an edge, the closer the scores of the two nodes connected by the edge becomes.

2.4 Graph-based active learning algorithms

An important categorization of graph-based active learning for applications is whether it is *interactive* or *non-interactive*. Here, interactive approaches use human labels during the learning process; they present a node for humans to label, and based on this label, the algorithms compute the next node to be presented to the humans. Thus, in interactive algorithms, human labeling and computations of the next node must run concurrently.

Non-interactive algorithms do not use human labels during the learning process. Given the entire graph, these algorithms sample important nodes for label propagation algorithms. Here, important nodes are the ones that minimize estimated classification error of label propagation when the nodes are labeled. Note that, unlike active learning used in the NLP community, non-interactive active learning algorithms exclude expert annotators' human labels from the protocol. While they exclude expert annotators, they are still regarded as active learning methods in the machine learning community since they try to choose such nodes that are beneficial for classification (Ji and Han, 2012; Gu and Han, 2012).

For educational purposes, *non-interactive* algorithms are preferred over *interactive* algorithms. The main drawback of interactive algorithms is that they must run concurrently with the human labeling. For our applications, this means that the vocabulary tests for vocabulary prediction must always be computerized. In contrast, non-interactive algorithms allow us to have vocabulary tests printed in the form of handouts, so we focus on non-interactive algorithms throughout this paper.

Compared with interactive algorithm studies, such as Zhu et al. (2003), graph-based non-interactive active learning algorithms have been introduced in recent years. There has been a seminal paper on non-interactive algorithms (Ji and Han, 2012). We used Gu and Han's algorithm because it reports higher accuracy for many tasks with competitive computation times over Ji and Han's algorithm (Gu and Han, 2012).

These active learning methods share two basic rules although their objective functions are different. First, these methods tend to select globally important nodes, also known as *hubs*. A notable example of global importance is the number of edges. Second, these methods tend to avoid sampling nodes that are heavily connected to previously sampled nodes. This is due to *cluster assumption*, the assumption that similar nodes should have similar labels, which suggests that it is redundant to select nodes close to previously sampled nodes; the labels of such nodes should be reliably predicted from the previously sampled nodes.

Gu and Han's algorithm, which is the algorithm we used, also follows these rules. In this algorithm, when considering the k-th sample, for every node i in the current set of not-yet-chosen nodes, a score $score(k, i)$ is calculated, and the node with the highest score is chosen. First, the score is de-

[1]While all elements of a non-negative definite matrix are not necessarily non-negative, we define all elements of $\mathbf{W}$ as non-negative here, following the definition of Zhou et al. (2004).

signed to be large if the i-th node is globally important. In the algorithm, the global importance of a node is measured by an eigenvalue decomposition of the normalized graph-Laplacian, $\mathbf{L}^{\text{norm}}$. Transformed from the graph's adjacency matrix, this matrix stores the graph's global information. Second, the score is designed to be smaller if the i-th node is close to one of the previously sampled nodes.

Score $score\,(k,i)$ is defined as follows. We perform eigenvalue decomposition beforehand. $\mathbf{L}^{\text{norm}}_{\mathbf{W}} = \mathbf{U}\Lambda\mathbf{U}^{\top}$, $\mathbf{u}_i$ is the transpose of the i-th row of $\mathbf{U}$, and λ_i is its corresponding eigenvalue.

$$score\,(k,i) \stackrel{\text{def}}{=} \frac{\left(\mathbf{H}_k^{-1}\mathbf{u}_i\right)^{\top}\mathbf{\Lambda}^{-1}\left(\mathbf{H}_k^{-1}\mathbf{u}_i\right)}{1+\mathbf{u}_i^{\top}\mathbf{H}_k^{-1}\mathbf{u}_i} \quad (2)$$

In Equation 2, $\mathbf{H}_k$ preserves information of the previous $k-1$ samples. First, $\mathbf{H}_0$ is a diagonal matrix whose i-th diagonal element is defined as $\frac{1}{(\mu\lambda_i+1)^2-1}$ where μ is a hyper-parameter. $\mathbf{H}_0$ weighs the score of globally important nodes through the eigenvalue decomposition. Second, $\mathbf{H}_k$ is updated such that the scores of the nodes distant from the previously taken samples are higher. The precise update formula of $\mathbf{H}_k$ follows. i_{k+1} is the index of the node sampled at $k+1$-th round. For the derivation of this formula, see Gu and Han (2012).

$$\mathbf{H}_{k+1}^{-1} = \mathbf{H}_k^{-1} - \frac{\left(\mathbf{H}_k^{-1}\mathbf{u}_{i_{k+1}}\right)\left(\mathbf{H}_k^{-1}\mathbf{u}_{i_{k+1}}\right)^{\top}}{1+\mathbf{u}_{i_{k+1}}^{\top}\mathbf{H}_k^{-1}\mathbf{u}_{i_{k+1}}} \quad (3)$$

Hyper-parameter μ determines how strong the cluster assumption should be; the larger the value, the more strongly the algorithm avoids selecting nodes near previously selected samples over the graph. Note that μ is inherited from the LLGC[2] algorithm (Zhou et al., 2004), i.e., the label propagation algorithm that Gu and Han's algorithm is based on. From the optimization viewpoint, μ determines the degree of penalization.

Remember that the $score$ has nothing to do with the LP scores described in §2.3. $score$ is used to choose nodes used for training in the graph-based non-interactive active learning. LP scores are later used for classification by label propagation algorithms that use the chosen training nodes. Throughout this paper, when we mean LP scores, we explicitly write "LP scores". All the other scores mean $score$.

[2]Learning with Local and Global Consistency.

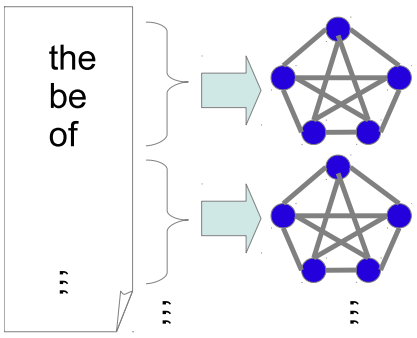

Figure 1: Converting frequency list into multiple-complete graph.

3 Formalizing heuristic word sampling as graph-based active learning

Figure 1 shows how to formalize a word frequency list into a multiple complete graph. The word frequency list is split into clusters, and each cluster forms a complete graph. Each node in a graph corresponds to a word. By gathering all the complete graphs, a multiple complete graph can be formed.

Multiple complete graph $\mathcal{G}_{T,n}$ is defined as a graph of T complete graphs, each of which consists of n nodes fully connected within the n nodes. An example of a multiple complete graph can be seen in Figure 2. We can define the $Tn \times Tn$ adjacency matrix for multiple complete graphs. $\mathbf{W}^{\text{complete}}_{\text{all}}$ is defined as follows:

$$\mathbf{W}^{\text{complete}}_{\text{all}} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{W}^{\text{complete}} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{W}^{\text{complete}} \end{pmatrix} \quad (4)$$

$$\mathbf{W}^{\text{complete}} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & 1 & \cdots & 1 & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ \vdots & 1 & \ddots & & \vdots \\ 1 & \vdots & & \ddots & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \quad (5)$$

We can see that $\mathbf{W}^{\text{complete}}_{\text{all}}$ is a block-diagonal matrix where each block is a $n \times n$ matrix, $\mathbf{W}^{\text{complete}}$.

Heuristic word sampling can be rewritten into non-interactive active learning on graphs. Suppose there are T groups, each of which has n words, and we want to sample n_0 words from each. In

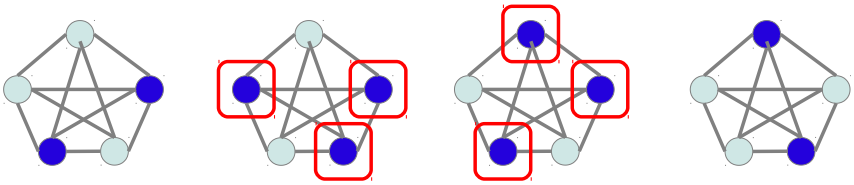

Figure 2: Example of multi-complete graph, where Theorem 3.1 holds true. Here, $T = 4$, $n = 5$, and $k = 10$; 10 light blue (light) nodes have already been sampled, and 10 blue (dark) nodes remain; the 11-th node is sampled uniformly randomly from the nodes within the red rectangles.

heuristic word sampling, for each group from T groups, n_0 words are sampled from the n words in the group uniformly randomly. Thus, there are Tn_0 words in total.

Since heuristic word sampling takes a node from each of the T groups, T concurrent sampling processes are involved. For simplicity, we further express the same sampling using only one sampling process from the entire graph as follows:

- For every round, we sample words uniformly randomly from the remaining words of the groups where the number of samples selected in previous rounds is least.

Figure 2 shows an example of this sampling process. Here, the second and third groups from the left are the groups in which the number of previously selected nodes is the least. This is because they have only two previously selected nodes, while the others have three. Thus, in the figure, the remaining words of the groups are the nodes with red rectangles. Randomly sampling one node from the nodes with red rectangles means sampling a node from the second or third group. We call the set of nodes in a graph from which samples will be taken in the next round a *seed pool*. Thus, in Figure 2, the set of nodes with red rectangles is the seed pool. Nodes that have already been sampled are taken out of the current seed pool.

Next, we more formally explain the seed pool concept. We start sampling nodes from a multiple complete graph via the algorithm presented by Gu and Han. The initial seed pool is set to all nodes in the graph, i.e., $\mathcal{V}$. We sample one node in each round; thus, $k \leq |\mathcal{V}|$ nodes are selected by the k-th round. Let $t \leq T$ be the index of the complete graph in the multiple complete graph. Then, the following theorem holds with ϵ being a small positive value that substitutes the 0 eigenvalues in the eigen decomposition.

Theorem 3.1 *Let* $0 < \epsilon < 1$ *and* $n \in \{2, 3, 4, \ldots\}$. *Then, among* T *complete graphs,* $k \mod T$ *complete graphs have* $\lfloor \frac{k}{T} \rfloor + 1$ *samples, and the remaining graphs have* $\lfloor \frac{k}{T} \rfloor$ *samples*[3]. *Moreover, the* $(k+1)$*-th sample is taken uniformly randomly from the remaining complete graphs.*

In Theorem 3.1, $\epsilon > 0$ is a substitute for the 0 eigenvalue of $\mathbf{L_W}$ [4]. Since ϵ is a substitute for the 0 eigenvalue, it is rational to assume $1 > \epsilon$. Also, remember that n is the number of nodes in one complete graph. The algorithm stops when $k = Tn_0 + 1$, i.e., at the $Tn_0 + 1$-th round when there are no remaining nodes to sample. Figure 2 shows an example of Theorem 3.1.

A proof of this theorem is presented in the supplementary material. Briefly, in a multiple complete graph, the score of a node depends only on the complete graph or the cluster that the node belongs to. Thus, we only have to consider one complete graph in which k is the number of nodes that have been already chosen. Then, mathematical induction proves that, within one complete graph, all the not-yet-chosen nodes have the same $score(k, i)$. Second, we have to show that the score always decreases by taking a sample, i.e., $score(k, i) > score(k+1, i)$. By a long but straightforward calculation, we can express $score(k, i)$ by using only μ, ϵ, n, and k. Then, by substituting the formula to $score(k, i)$, we obtain $score(k, i) - score(k+1, i) > 0$.

4 Extending Graphs

In the previous section, we explained how to formalize heuristic word sampling as active learning on multiple complete graphs. This formaliza-

[3] Here, both k and T are non-negative integers. Thus, $k\%T$ denotes the remainder of the division of k by T, and $\lfloor \frac{k}{T} \rfloor$ is the quotient of the division.

[4] In Gu and Han's algorithm, they substitute the 0 eigenvalue with a small positive value ϵ, and they set $\epsilon = 10^{-6}$.

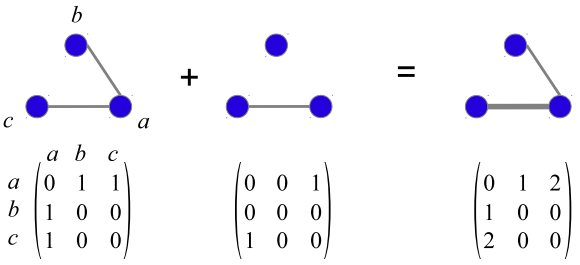

Figure 3: Example of merging two graphs.

tion can lead to better active learning by extending these graphs. In this section, we describe such graph extensions.

We extend graphs by *merging* graphs. Figure 3 shows how to merge graphs. We define "merging" two weighted graphs as creating a weighted graph whose adjacency matrix is the sum of the two adjacency matrices of the two weighted graphs. This suggests that an edge of the merged graph is simply the sum of the corresponding edges of the two weighted graphs.

The merged graph is expected to inherit the characteristics of its original graphs. Thus, applying graph-based active learning to the merged graph is expected to sample nodes in accordance with the characteristics of its original graphs. For example, if we merge a graph representing domain-specific relations and a multiple complete graph representing difficulty grouping of words, active learning from the resulting merged graph is expected to sample words considering both domain specificity and difficulty grouping of words.

For another example, suppose we merge two multiple complete graphs created from frequency lists from two different corpora. Then, active learning from the resulting merged graph is expected to sample words taking into account frequency lists from both corpora.

5 Evaluation

We evaluate our proposed method both quantitatively and qualitatively. In the quantitative evaluation, we measure the prediction accuracy of graphs. Note that the heuristic word sampling method is identical to using Gu and Han's algorithm with a multiple complete graph; however, our proposed graphs have enriched relations between words. In the qualitative evaluation, we explain in detail what words are appropriate as training examples for vocabulary prediction by presenting sampled examples.

5.1 Quantitative evaluation

To evaluate the accuracy of vocabulary prediction, we used the dataset that Ehara et al. (2010) and Ehara et al. (2012) used. This dataset was gleaned from questionnaires answered by 15 English as a second language (ESL) learners. Every learner was asked to answer how well he/she knew 11,999 English words. The data was collected in January 2009. One learner was unpaid, whereas the other 15 learners were paid. We used the data from the 15 paid learners since the data from the unpaid learner was noisy. Most of the learners were native Japanese speakers and graduate students. Because most of the learners in this dataset were native Japanese speakers, words from SVL 12,000 (SPACE ALC Inc., 1998) were used for the learners in this dataset. Note that SVL 12,000 is a collection of 12,000 words that are deemed important for Japanese learners of English, as judged by native English teachers.

Next, we required frequency lists for the words that appeared in the dataset. To create frequency lists, lemmatization is important because the number of word types depends on the method used to lemmatize the words. Note that in the field of vocabulary measurement, lemmatization is mainly performed by ignoring conjugation (Nation and Beglar, 2007). Lemmatizing the dataset resulted in a word list of 8,463 words. We adjusted the size of the word list to a round 8,000 by removing 463 randomly chosen words. Note that all constituent words were labeled by the 15 ESL learners.

We created the following four graphs by spanning edges among the $8,000$ words.

BNC multi-complete This graph corresponds to heuristic word sampling and served as our baseline. It is a multiple complete graph comprising eight complete graphs, each of which consisted of 1,000 words based on the sorted frequency list from the British National Corpus (BNC). We chose the BNC because the method presented by Nation and Beglar was based on it (Nation and Beglar, 2007). Note that all edge weights are set to 1.

BNC+domain To form this graph, edges representing domain specificity are added to the "BNC multi-complete" graph. For domain specificity, we used domain information

from WordNet 3.0.[5] First, we extracted 102 domain-specific words under the "computer" domain among the 8,000 words and created a complete graph consisting of these domain-specific words. The edge weights of the complete graph were set to 1. Next, we simply *merged*[6] the complete graph consisting of the domain-specific words with the "BNC multi-complete" graph.

BNC+COCA In addition to the "BNC multi-complete" graph, edges based on another corpus, the Corpus of Contemporary American English (COCA), were introduced. We first created the COCA multi-complete graph, a multiple complete graph consisting of eight complete graphs, each of which consisted of 1,000 words based on the sorted frequency list using COCA. The edge weights of the COCA multi-complete graph were set to 1. Next, we merged the BNC multi-complete and COCA multi-complete graphs to form the "BNC + COCA graph".

BNC+domain+COCA This graph is the graph produced by merging the "BNC + domain" and "BNC + COCA" graphs.

Note that our experiment setting differed from the usual label propagation setting used for semi-supervised learning because the purpose of our task differed. In the usual label propagation setting, the "test" nodes (data) are prepared separately from the training nodes to determine how accurately the algorithm can classify *forthcoming* or *unseen* nodes. However, in our setting, there were no such forthcoming words. Of course, there will always be words that do not emerge, even in a large corpus; however, such rare words are too difficult for language learners to identify, and many are proper nouns, which are not helpful for measuring the vocabulary of second language learners.

Therefore, our focus here is to measure how well the learners know a fixed set of words, that is, the given 8,000 words. Even if an algorithm can achieve high accuracy for words outside this fixed set, we have no way of evaluating it using the pooled annotations. Here, we want to measure, from a fixed number of samples (e.g., 50), how accurately an algorithm can predict a learner's vocabulary for the entire 8,000 words. Thus, we define accuracy to be the number of words that each algorithm finds correctly divided by the vocabulary size. We set hyper-parameter μ to 0.01 as Gu and Han (2012) did. Note that this hyper-parameter is reportedly not sensitive to accuracy (Zhou et al., 2011).

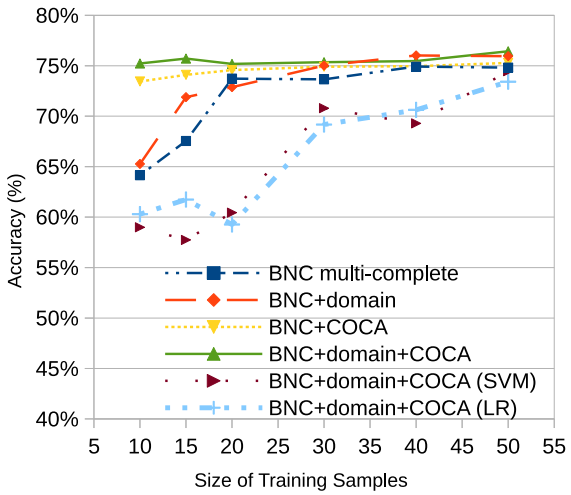

Figure 4: Results of our quantitative experiments. Vertical axis denotes accuracy, and horizontal axis shows number of samples, i.e., training words.

Figure 4 and Table 1 show the results of the experiment over the different datasets. The vertical axis in the figure denotes accuracy, whereas the horizontal axis denotes the number of samples, i.e., training words. Note that the accuracy is averaged over 15 learners and that LLGC is used for classification unless otherwise specified. For example, "BNC multi-complete" indicates that samples taken from the BNC multi-complete graph are used for training, and LLGC is used for classification. Note that "BNC + domain + COCA (SVM)" uses a support vector machine (SVM) for classification, and "BNC + domain + COCA (LR)" uses logistic regression (LR) for classification. Among many supervised machine learning methods, we chose SVM and LR because SVM is widely used in the NLP community, and LR was used for theoretical reasons (Ehara et al., 2012; Ehara et al., 2013).

SVM and LR require features of a word for classification while LLGC requires a weighted graph of words. Since the graph "BNC+domain+COCA" is made from three features, namely the word frequencies of BNC

[5]We used the NLTK toolkit http://nltk.org/ to extract the domain information.

[6]Definition of how to *merge* two graphs is in §4.

Table 1: Results of our quantitative experiments. LLGC is used for classification unless otherwise specified. Bold letters indicate top accuracy. Asterisks (*) indicate that values are statistically significant against baseline, heuristic sampling, i.e., "BNC multi-complete" (using sign test $p < 0.01$).

	10	15	20	30	40	50
BNC multi-complete	64.15 (%)	67.54	73.73	73.66	74.92	74.82
BNC+domain	65.27	71.88	72.88	75.02	**76.03** *	75.95
BNC+COCA	73.45	74.10	74.57	74.90	74.96	75.29
BNC+domain+COCA	**75.23** *	**75.71** *	**75.18** *	**75.35** *	75.47	**76.44** *
BNC+domain+COCA (SVM)	58.99	57.74	60.44	70.79	69.29	74.46
BNC+domain+COCA (LR)	60.29	61.74	59.27	69.17	70.63	73.42

and COCA corpora and whether a word is in the computer domain, we used these features for the features of SVM and LR in this experiment for a fair comparison. When using word frequencies for features, we used the logarithm of raw frequencies since it is reported to work well (Ehara et al., 2013). SVM and LR are also known to heavily depend on a hyper-parameter called C, which determines the strength of regularization. We tried $C = 0.1, 0.5, 1.0, 2.0, 5.0, 10.0, 20.0, 50.0,$ and 100.0 for each of SVM and LR where the size of training data is 50 and chose the C value that performs best. As a result, we set $C = 5.0$ for SVM and $C = 50.0$ for LR. Note that this setting is advantageous for SVM and LR compared to LLGC because the hyper-parameters of SVM and LR are tuned while LLGC's hyper-parameter remains untuned. For the implementation of SVM and LR, we used the "scikit-learn" package in Python [7].

We first observed that our proposed methods constantly outperform the baseline, heuristic word sampling, i.e., "BNC multi-complete" in Table 1. This indicates that we successfully obtained better accuracy by formalizing heuristic word sampling as active learning and extending graphs. In Table 1, the accuracy of the top-ranked methods (shown using **bold** letters) is statistically significantly better than the accuracy of "BNC multi-complete" (using the sign test $p < 0.01$).

We then observed that "BNC multi-complete" and "BNC + domain" show competitive accuracy with sample sizes from 10 to 20; furthermore, "BNC + domain" is slightly better than "BNC multi-complete" with sample sizes ranging from 30 to 50 (statistically significant $p < 0.01$ using sign test). Next, we note that there is a trade-off between domain and word frequency when choosing samples. More specifically, if we select too many words from the domain, the measurement of the general English ability of learners can be inaccurate; conversely, if we select too many words from the corpus-based word frequency list, while the general English ability of learners is accurately measured, we may obtain no information on the learner's vocabulary for the targeted domain. The competitive or slightly better accuracy of "BNC + domain" over "BNC multi-complete" shows that "BNC + domain" could successfully integrate domain information into the frequency-based groups without deteriorating measurements of general English ability.

We also observe that "BNC + COCA" greatly outperforms "BNC multi-complete" when the number of samples is 10. This shows that the integration of the two corpora, BNC and COCA (i.e., "BNC + COCA"), successfully increases the accuracy when there are only a small number of samples.

"BNC + domain + COCA" achieves the best accuracy of all the graphs except when the number of samples is 40. This indicates that the domain information and the information from the COCA corpus helped one another to improve the accuracy because "BNC + domain" and "BNC + COCA" introduce different types of domain information into "BNC multi-complete."

Finally, we observe that "BNC + domain + COCA (SVM)" and "BNC + domain + COCA (LR)" perform worse than LLGC over the same dataset for all sample sizes, particularly when the size of the training data is small. Since LLGC is a semi-supervised classifier while SVM and LR are not, SVM and LR perform poorly for small amounts of training data. This result shows that LLGC is appropriate for this task compared to SVM because, in this task, an increase in the size

[7] http://scikit-learn.org/stable/

Table 2: Computer-related samples in top 30 samples.

Name	Num. of Samples	Examples
BNC multi-complete	0	-
BNC+domain	5	input, client, field, background, register
BNC+COCA	0	-
BNC+domain +COCA	3	drive, client, command

of training data directly leads to an increased burden on the human learners.

5.2 Qualitative evaluation

In this subsection, we qualitatively evaluate our results to determine the types of nodes that are sampled when domain specificity is introduced. Specifically, we evaluate what words are selected as samples in the "BNC + domain" graph.

As noted above, in the "BNC + domain" graph, the computer science domain is introduced into "BNC multi-complete" to measure learners' vocabulary with a specific emphasis on the computer science domain. Thus, it is desirable that some words in the computer science domain are sampled from the "BNC + domain" graph; otherwise, we need to predict the learners' vocabulary for the computer science domain from general words rather than those in the computer science domain, which is extremely difficult.

Table 2 shows the number of words in the computer science domain sampled in the first 30 samples. Note that only "BNC + domain" and "BNC + domain + COCA" select samples from the computer science domain. This indicates that in the other two methods, to measure vocabulary with an emphasis on the computer science domain, we need to predict learners' vocabulary from the general words, which is almost impossible with only 30 samples. Furthermore, it is interesting to note that "BNC + domain" and "BNC + domain + COCA" select different samples from the computer science domain, except for the word "client," although originally the same computer science domain wordlist was introduced to both graphs.

Since "BNC + domain" achieves competitive or slightly better accuracy than "BNC multi-complete" in the quantitative analysis and the qualitative analysis, we conclude that our method can successfully introduce domain specificity into the sampling methodology without reducing accuracy.

6 Conclusion

In this study, we propose a novel sampling framework that measures the vocabulary of second language learners. We call existing sampling methods heuristic sampling. This approach to sampling ranks words from a single corpus by frequency and creates groups of 1,000 words. Next, tens of words are sampled from each group. This method assumes that the relative difficulty of all 1,000 words is the same.

In this paper, we introduce a novel sampling method by showing that the existing heuristic sampling approach is simply a special case of a graph-based active learning algorithm by Gu and Han (2012) applied to a special graph. We also propose a method to extend this graph to enable us to handle domain specificity of words and multiple corpora, which are difficult or impossible to handle using current methods.

We evaluate our method both quantitatively and qualitatively. In our quantitative evaluation, the proposed method achieves higher prediction accuracy compared with the current approach to vocabulary prediction. This suggests that our proposed method can successfully make use of domain specificity and multiple corpora for predicting vocabulary. In our qualitative evaluation, we examine the words sampled by our proposed method and observe that targeted domain-specific words are successfully sampled.

For our future work, because the graph used in this paper was constructed manually, we plan to automatically create a graph suitable for active learning and classification. There are several algorithms that create graphs from feature-based representations of words, but these have never been used for active learning of this task.

Acknowledgments

This work was supported by the Grant-in-Aid for JSPS Fellows (JSPS KAKENHI Grant Number 12J09575).

References

Yo Ehara, Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa. 2010. Personalized reading support for second-language web documents by collective intelligence. In *Proceedings of the 15th international conference on Intelligent user interfaces (IUI 2010)*, pages 51–60, Hong Kong, China. ACM.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2012. Mining words in the minds of second language learners: learner-specific word difficulty. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, Mumbai, India, December.

Yo Ehara, Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa. 2013. Personalized reading support for second-language web documents. *ACM Transactions on Intelligent Systems and Technology*, 4(2).

Quanquan Gu and Jiawei Han. 2012. Towards active learning on graphs: An error bound minimization approach. In *Proceedings of the IEEE International Conference on Data Mining (ICDM) 2012*.

Ming Ji and Jiawei Han. 2012. A variance minimization criterion to active learning on graphs. In *Proceedings of the 15th international conference on Artificial Intelligence and Statistics (AISTATS)*.

Batia Laufer and Paul Nation. 1999. A vocabulary-size test of controlled productive ability. *Language testing*, 16(1):33–51.

Paul Meara and Barbara Buxton. 1987. An alternative to multiple choice vocabulary tests. *Language Testing*, 4(2):142–154.

Paul Nation and David Beglar. 2007. A vocabulary size test. *The Language Teacher*, 31(7):9–13.

SPACE ALC Inc. 1998. Standard vocabulary list 12,000.

Dengyong Zhou, Oliver Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *Proceedings in 18th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 321–328.

Xueyuan Zhou, Mikhail Belkin, and Nathan Srebro. 2011. An iterated graph laplacian approach for ranking on manifolds. In *Proceedings of 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 877–885.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. 2003. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*.