Domain Independent Vocabulary Generation and Its Use in Category-based Small Footprint Language Model

Kwang-Ho KIM, Ji-Hwan KIM
Department of Computer Science and Engineering, Sogang University
1 Sinsu-dong, Mapo-gu, Seoul 121-74, Korea
{kimkwangho, kimjihwan}@sogang.ac.kr

***Abstract*—The work in this paper pertains to domain independent vocabulary generation and its use in category-based small footprint Language Model (LM). Two major constraints of the conventional LMs in the embedded environment are memory capacity limitation and data sparsity for the domain-specific application. This data sparsity adversely affects vocabulary coverage and LM performance.**

To overcome these constraints, we define a set of domain independent categories using a Part-Of-Speech (POS) tagged corpus. Also, we generate a domain independent vocabulary based on this set using the corpus and knowledge base. Then, we propose a mathematical framework for a category-based LM using this set. In this LM, one word can be assigned assign multiple categories. In order to reduce its memory requirements, we propose a tree-based data structure. In addition, we determine the history length of a category n-gram, and the independent assumption applying to a category history generation.

The proposed vocabulary generation method illustrates at least 13.68% relative improvement in coverage for a SMS text corpus, where data are sparse due to the difficulties in data collection. The proposed category-based LM requires only 215KB which is 55% and 13% compared to the conventional category-based LM and the word-based LM, respectively. It successively improves the performance, achieving 54.9% and 60.6% perplexity reduction compared to the conventional category-based LM and the word-based LM in terms of normalized perplexity.

***Index Terms*—Natural language processing, Speech recognition**

I. INTRODUCTION

We have seen some remarkable progress in spoken language processing technology so far, but because of the limitation of memory capacity and computation power, its general usage in the embedded environment has remained stagnate at the stage of recognizing isolated words. Although the fast spread of smartphones in recent months has made Google, Nuance, and Vlingo release a few smartphone applications, which recognize continuous speech, smartphones only extract features and these applications feed the recognition results from various servers, in the so called cloud.

Speech recognition aims to find the most likely word sequence of $\hat{W}$ which maximizes the multiplication of $P(W|O)$ and $P(W)$ which are provided by an Acoustic Model (AM) and a Language Model (LM), respectively[1]. For LMs, a word is the most fundamental observation unit. It is necessary to choose a set of words, before building the LM. This set of words is called the vocabulary.

The complexity of an LM is controlled by the number of words in the vocabulary, the history length, the total number of units in the representation of a history, and the amount of training corpus. However, the complexity of an AM depends only on the definition of the base recognition unit (e.g. triphone, phoneme) and the topology of each unit. As a result, the amount of memory required for an AM is determined as a fixed value before the domain of an application and the amount of training corpus are determined. According to an empirical comparison of several well-known speech recognition systems such as HTK [2] and Sphinx [3], it was reported that LMs consume more than 3.5GB which are approximately 98% of the total memory requirements of their corresponding speech recognizers, while they are trained in employing 222M word corpus with 65K word vocabulary [4].

In spite of the limitation of memory capacity, if the vocabulary and form of sentences are limited as in the case of Short Message Service (SMS), the implementation of an LM becomes feasible in the embedded environment through limiting the size of the vocabulary and reducing the memory footprint of an LM. Another aspect to be considered for this LM implementation is the difficulty in gathering the LM training corpus. As services in the embedded environment involve numerous personal information of different properties such as in SMS, GPS coordinates, and names in a phonebook, it costs a great deal to gather the corpus under users' contents.

The work in this paper pertains to domain independent vocabulary generation and its use in category-based small footprint LM. Two major constraints of conventional LMs in the embedded environment are memory capacity limitation and data sparsity for the domain-specific application. This data sparsity adversely affects vocabulary coverage and LM performance. To overcome these constraints, we define a set of domain independent categories using a POS tagged corpus. Also, we generate a domain independent vocabulary based on this set using the corpus and knowledge base. Then, we propose a mathematical framework for a category-based LM using this set. A category is assigned to a group of words which have same syntactic functions or similar semantic meanings. One example of categories is POS. 'A' and 'AN' are chosen as

This work was supported by the Sogang University Research Grant of 200810032.01.

Corresponding author : Ji-Hwan Kim

Digital Object Identifier 10.4316/AECE.2011.01013

an example. The both are classified by POS as an 'Indefinite Article', and assigned to the same word category.

This paper consists of six chapters. In Chapter II, the related studies to vocabulary generation and LM implementation are explained. In Chapter III, a domain independent vocabulary generation is presented. In Chapter IV, a category-based small footprint LM is proposed. In Chapter V, our proposed vocabulary generation and category-based LM are evaluated. This paper concludes in Chapter VI.

II. Previous Work

Related works to vocabulary generation are explained in Section II-A. Works regarding LM implementation are described in Section II-B.

A. Vocabulary Generation

Previous approaches in the generation of vocabulary use the frequency of each word used in the corpus. If coverage is defined as the probability of the word existing in the vocabulary, the vocabulary, which is optimized for the corpus in terms of coverage, is generated according to the frequency of each word used in the corpus. If the corpus is given, the vocabulary optimized for the corpus is generated according to the frequency of each word used in the corpus. TABLE I shows the total number of words in the corpus, the total number of words in the vocabulary and the coverages were measured for an English newspaper text corpus using the above method [5]. In this domain, the maximum coverages with 5K, 20K and 65K words were measured at 90.6%, 97.5% and 99.6%, respectively.

TABLE I. Coverage According to Size of Vocabulary Optimized for an English Newspaper Corpus [5]

Total No. of Words in Corpus	Total No. of Words in Vocabulary	Coverage According to Vocabulary Size		
		5K	20K	65K
37.2M	165K	90.6%	97.5%	99.6%

In order to determine the size of the necessary corpus when the amount of vocabulary is fixed and the vocabulary is generated according to the frequency of usage for each word in the corpus, the test for the change of coverage according to the size of the corpus is carried out [6]. In this test, coverages were measured by increasing the quantity of newspaper text corpus by 5M when the sizes of vocabulary were fixed at 20K words, 40K words, and 60K words. As the size of corpus gets bigger, the coverage also becomes higher, but once the size is over 30M words, the improvement of coverage becomes scarce. The coverages were measured at about 96%, 98%, and 98.5% when the sizes of vocabulary were 20K, 40K and 60K words respectively.

Services in the embedded environment such as SMS involve many personal messages. Thus, it is very difficult to collect such amount of corpus as is required to measure the exact frequency of the words as described in [6]. If it is possible to effectively estimate the word frequencies for the SMS from an easily collectable corpus in other domains as in news material, then we could decide the most suitable vocabulary according to word frequencies of the corpus from the corresponding domain. However, the coverage of the same sized vocabulary for various domains shows great difference. According to the test results of [5] summarized in TABLE I, the coverage of 5K word vocabulary for the newspaper text corpus was measured at about 90%. However, the coverage of the same size vocabulary for the spoken part of the British National corpus was at 96.9% [7]. In the case of CANCODE corpus that was transcribed from the recorded conversations from UK and Ireland with 5M words, the coverage of the same size vocabulary was at 96.1% [7]. In written language, there is much difference in the coverage of same size vocabularies for various domains. According to [8], for the most common 2K word lists from the written corpora, the coverage of academic references, newspapers, magazines, and novels were measured at 78.1%, 80.3%, 82.9% and 87.4% respectively.

B. Language Model Implementation

The standard approach in LM is the word-based *n*-gram [4]. If all histories with the same most recent *n*-1 words are treated as equivalently, the word-based *n*-gram probability is approximated as in (1):

$$P(w_i \mid w_1,...,w_{i-1}) \approx P(w_i \mid w_{i-n+1},...,w_{i-1}) \qquad (1)$$

The simplicity of the word-based *n*-gram LM has both its greatest strength and weakness. On the positive side, the model's simplicity means that the models are easy and efficient to train, even though corpora of hundreds of millions of words are used. The models can be constructed on a large amount of real data. This large amount of real data is also necessary for the generation of vocabulary with reasonable coverage. The negative sides are its memory requirement and data sparsity. Even for a trigram with a 60k word vocabulary, it must be able to produce probabilities for 2.2×10^{14} different word sequences. The common scheme for alleviating the problems of data sparsity is to use a less descriptive model, such as a back-off model [9]. In this scheme, the probability of hypothesized word sequences, which do not occur in the model, is estimated by backing off to a lower order *n*-gram model.

The tree-based statistical language model was implemented based on a decision binary tree [10]. In this implementation, the LM probability of the next word is estimated based on the query results from the decision tree for its history of the previous 20 words. The disadvantage of this LM is that it lacks domain portability. When a tree is constructed, once the best splitting rule is found, the parent node is split and then the same procedure is repeated for each child node. This algorithm proceeds recursively until splitting is no longer possible. This tree continues splitting until it classifies its training data with 100% accuracy. As a result, this tree fits the training domain very well, however it does not guarantee the best performance for the test domain because this tree is over fitted to the training domain.

N-gram LM's major disadvantage is that it uses the immediate history only. To reinforce and complement against this disadvantage of the *n*-gram LM, trigger-based LM was introduced [11]. This model changes its estimates with consideration to the result of a history pattern seen from some of the text. In order to estimate the probability of a word, this model relies on the history of the documents observed after its original model building. The major advantage of this model is in its incremental value. On the

other hand, it demands a heavy load of computation.

Variable-length sequence model was presented in [12]. An *n*-gram LM looks up a static length of history in estimating the probability of a word. In contrast to the *n*-gram model, multi-gram histories are being examined in variable-length sequence model, and as a result, the performance of LM is improved. However, this model has a weakness in increased computation complexity, as it has a tendency to generate numerous combinations from a history.

Category-based LM was investigated in [13]. In category-based LM, a history is represented as a sequence of word categories. The use of the word category has some clear advantages. Firstly, the introduction of categories reduces the number of model parameters, because the number of different categories is considerably smaller than the number of words in the vocabulary. Secondly, word categories are used to predict smoothed probabilities of the occurrence of word sequences, which may be seldom or never have been observed in the training corpus. This LM is known as a robust LM to sparse data set and produces better prediction for word tuples not presented in the training set [14]. This improves the robustness of the probability estimates for a specific training corpus and, at the same time, positively affects the training data requirements. However, to the best of our knowledge, previous studies on the category-based LM have focused only on performance improvement, and not on memory requirement improvement.

A number of previous studies related to the category-based LM have investigated automatic generation of word categories and its use in the implementation of the category-based LM [15][16]. This auto clustering method aims to maximize the log likelihood of the training corpus. The clustering method starts with some initial assignments. For example, assign the most frequent *k*-1 words to their own categories and the remaining words to the other category, when *k* categories are automatically generated. This method evaluates the change in training corpus probability by moving every word in the vocabulary from its current category to every other possible category. This process is repeated until some minimum change threshold has been reached or a chosen number of iterations have been performed.

The main advantage of this category-based LM is that all the category generation processes are automatic. The generated word categories guarantee the maximum log likelihood over a certain training corpus.

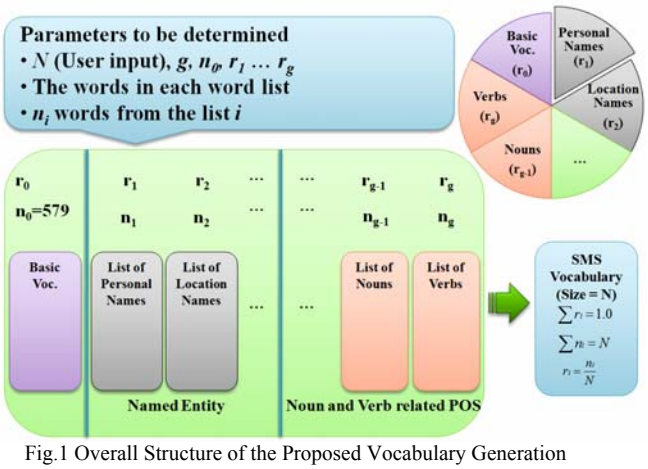

Fig.1 Overall Structure of the Proposed Vocabulary Generation

However, this category-based LM has two problems. First, words in a category often look unreasonable, because the categories are generated based on the maximization of the log likelihood, not on a syntactic function or a semantic similarity. When a new domain is introduced, it is impossible to assign a category to an out-of-vocabulary word, which is not listed in the existing vocabulary. Second, every word must be assigned to only one category. Considering the fact that a word has multiple syntactic or semantic functions, multiple categories should be allowed to be mapped to a single word.

III. Domain Independent Vocabulary Generation

Fig. 1 shows the overall structure of the proposed domain-independent vocabulary generation. If a user decides the size of vocabulary *N*, the basic vocabulary including n_0 words are already classified as belonging to the vocabulary. The basic vocabulary is a group of words that should be included in a vocabulary regardless of domain and the size of vocabulary. The n_0 basic vocabulary words are chosen by investigating LOB (Lancaster-Oslo/Bergen) corpus [17]. The LOB corpus is one of the most representative POS tagged corpora.

Then, there are *g* POSs in which the number of their corresponding words is uncountable. The examples of these POSs include named entity related POS, noun and verb. Since the number of corresponding words is uncountable, it is impossible to investigate whole of the words one-by-one, so we need to design a method to choose n_i words to put into the vocabulary from each POS. These n_i words constitute the rest of the vocabulary (in total N-n_0).

In Section III-A, we investigate 152 POS tags defined in a POS tagged corpus and word-POS tag pairs. We analyze all words paired with 101 among 152 POS tags and decide on a set of words which have to be included in vocabularies of any size. We define these words as basic vocabulary. In Section III-B, we describe a domain corpus independent word inclusion method for the other 51 POSs using knowledge base.

A. Basic Vocabulary Generation using POS Classification

In this research, we use LOB corpus. This is a British English text corpus including 1.16M words. TABLE II shows the total number of words, the size of vocabulary, and the number of POSs of the corpus.

TABLE II. Statistics of LOB Corpus

Total No. of Words	Size of Vocabulary	No. of POSs
1,157,278	56,174	152

The 152 POSs are defined in LOB corpus. All words in the corpus are paired with one of these 152 POSs as in the form of *'Word-POS'*. An excerpt is as follows:

she_PP3A had_HVD been_BEN in_IN mental_JJ hospitals _NNS since_IN 1944_CD ._.

In the above example, the POSs corresponding to the words 'she', 'in', and 'hospital' are 'PP3A' (singular form of third

person pronoun), 'IN' (preposition), and 'NNS' (plural form of a common noun) respectively.

For the 152 POSs of LOB corpus, according to how the corresponding words to POS belong to vocabulary, the POS is largely divided into 4 groups. TABLE III explains these 4 groups.

TABLE III. CLASSIFICATION OF POS ACCORDING TO THE METHOD OF HOW THE CORRESPONDING WORDS TO POS BELONG TO VOCABULARY AND ITS EXAMPLE (EXAMPLE FORMAT: POS TAG - CORRESPONDING WORD)

POS Group	Description
Group1 (No. of POS: 76)	All the words belonging to POS are included in the vocabulary regardless of the size of vocabulary. (e.g. BE-be, BEM-am, EX-there, HV-have, HVD-had)
Group 2 (No. of POS: 11)	These are grammatically important POS and the corresponding words according to POS are not many. Therefore the professionals decide whether they are included in the vocabulary regardless of the size of vocabulary. (e.g. CD-two, three, hundred, OD-10th, 15th, tenth, twenty-first)
Group 3 (No. of POS: 51)	The number of corresponding words is uncountable; therefore the whole of words cannot be investigated manually. The examples of these POSs include named entity related POSs, nouns and verbs. (e.g. NN-breakfast, NP-Henry, VB-generate)
Group 4 (No. of POS: 14)	These words do not have any possibility of being included in the vocabulary (e.g. comma, dash, semicolon)

Group 1 consists of POSs where all the words belonging to POS are included in the vocabulary regardless of the size of vocabulary. For example, the verbs 'be', auxiliary verbs, and the conjunctions are included. Group 2 consists of grammatically important POSs to which the corresponding words are not many. Therefore for all of the corresponding words, it is possible for the professionals to decide whether they are included in vocabulary regardless of the size of vocabulary. Their examples are cardinal numbers (e.g. one, two, hundred) and ordinal numbers (e.g. first, second). For all the words belonging to group 1 and group 2, professionals in natural language processing decide whether they belong to the vocabulary, and classify the basic vocabulary words which are in vocabulary included words regardless of the size of the vocabulary. As a result of this classification, it was decided that 579 words should be included in all the vocabulary regardless of its size. Thus, n_0 is decided as 579.

From group 3, the number of words added to the vocabulary is decided as the rest of the vocabulary, after including basic vocabulary words of group 1 and group 2 (579). That is N-n_0. There are g POSs in which the number of their corresponding words is uncountable. The examples of these POSs include named entity related POS, noun and verb. The number of POSs belonging to group 3 is altogether 51, and that number becomes g. The generation rates between POSs (r_i) are estimated by composition ratios of corresponding POS from LOB corpus. Since the number of words corresponding to the g POSs is uncountable, it is impossible to investigate whole of the words one-by-one, so we need to design a method to choose n_i words to put into the vocabulary from the POS. Each word in a list is sorted in descending order according to its relative importance. This relative importance is determined by using knowledge base. Section III-B will describe the details of how this relative importance is determined by using knowledge base. TABLE IV shows the classification results of the 152 POSs defined in LOB corpus.

TABLE IV. CLASSIFICATION RESULT OF POSS DEFINED IN LOB CORPUS

POS Group	Description
Group1	ABL, ABN, ABX, AP, AP", AP$, APS, APS$, AT, ATI, BE, BED, BEDZ, BEG, BEM, BEN, BER, BEZ, CC, CC", CS, CS", DO, DOD, DOZ, DT, DT$, DTI, DTS, DTX, EX, HV, HVD, HVG, HVN, HVZ, IN", MD, PN, PN", PN$, PP$, PP$$, PP1A, PP1AS, PP1O, PP1OS, PP2, PP3, PP3A, PP3AS, PP3O, PP3OS, PPL, PPLS, PPLS", QL, QLP, RB$, RI, RN, RP, TO, TO", WDT, WDT", WDTR, WP, WP$, WP$R, WPA, WPO, WPOR, WPR, WRB, IN
Group 2	CD, CD$, CD-CD, CD1, CD1$,CD1S, CDS, OD, OD$, XNOT, ZZ
Group 3	RB", RBR, RBT, UH, JJ, JJ", JJB, JJB", JJR, JJR", JJT, JJT", JNP, NC, NN, NN", NN$, NNP, NNP$, NNPS, NNPS$, NNS, NNS", NNS$, NNU, NNU", NNUS, NP, NP$, NPL ,NPL$, NPLS, NPLS$, NPS, NPS$, NPT, NPT", NPT$, NPTS, NPTS$, NR, NR$, NRS, NRS$, RB, VB, VB", VBD, VBG, VBN, VBZ
Group 4	&FO, &FW, !, (,), *', **', *_, ,, ., ..., :, ;, ?

B. The Estimation of Word Relative Importance Using Knowledge Base

In this section, the estimation of word relative importance for nouns, verbs, and named entities is explained. The 51 POS tags in group 3 are mainly categorized with noun-related, named entity related and verb-related POSs. In Section III-B-1), the vocabulary inclusion method for noun-related POSs is described. Synonym groups of words in noun-related POSs are generated using Wordnet and the relative importance between these synonym groups is determined by Google search. In Section III-B-2), the vocabulary inclusion method for verb-related POSs is explained. Verbs are grouped according to lemma considering variant word forms. The vocabulary inclusion of each verb group is determined by pre-analyzed statistics for corresponding verb group. In Section III-B-3) the method for named entity using Google search is discussed.

1) Creating Noun Synonyms Using Wordnet

WordNet [18] consists of synsets which are interlinked by conceptual semantic and lexical relations. The POSs used in WordNet are nouns, verbs, adjectives and adverbs. Wordnet classifies synsets differently according to the type of POS. For nouns Wordnet provides the synsets such as hypernym, hyponym, synonym, holonym and meronym. The synonym group is a set of words having the same hypernym. TABLE V shows the total number of words and synonym groups for each POS.

TABLE V. STATISTICS OF WORDNET

POS	No. of Words	No. of Synonyms
Noun	117,798	82,115
Verb	11,529	13,767
Adjective	21,479	18,156
Adverb	4,481	3,621
Total	155,287	117,659

The method of vocabulary inclusion for words corresponding to noun-related POSs is as follows: Firstly, the noun synonym group is generated for each noun using Wordnet. For example, when a synonym group for the noun 'pear' is queried, Wordnet provides 'edible fruit' which is a hypernym for 'pear'. Then noun synonym group for 'pear' is generated including 'apple', 'berry', 'pineapple' and 'banana' which are hyponyms of 'edible fruit'. These words can replace 'pear' in any sentences in which 'pear' appears.

In this paper, noun synonym groups are generated using Wordnet, for each noun-related word listed in the conventional vocabulary (depending on the word, plural noun synonym groups can be generated). As a result, the synonym groups belonging to every noun-related word are created using Wordnet. Each synonym group is named as the corresponding hypernym. Hyponyms having the same hypernym are included in the same synonym group. Secondly, the relative importance between noun-related words according to synonym group in vocabulary is determined by knowledge base. In order to obtain the importance of noun-related words in vocabulary from the knowledge base, it is necessary to extract the desired importance from the query result after querying each word to the knowledge base. In this paper, the importance using knowledge base is estimated by the frequency of documents appearing in Google search for each word. When each word requests Google search, Google provides the number of documents related to that word. The Google search contains billions of documents, so that it is possible to obtain enough statistical values related to the required relative importance between words in vocabulary. In this proposed method, the initial importance of noun-related words in the conventional vocabulary is configured using Google search.

Starting from the noun word with the highest importance, synonyms for that word are generated by using Wordnet. Then the relative importance of these synonyms is obtained from Google search. According to relative importance, the words in each synonym group are sorted in descending order. The relative importance of the word in the sorted list is compared with the lowest importance in the current noun word list. If the importance of the word in the sorted list is greater than that of the compared word in current noun word list, these two words are exchanged, and as a result a synonym word is included in the vocabulary. This procedure is continued to the next word in the synonym groups until the above statement is satisfied. After finishing the determination of vocabulary inclusion or exclusion to all of the words in a synonym group, we repeat this determination to all of the words in the next synonym group.

2) Estimation of Relative Importance between Verbs by Lemma

Verbs change forms according to tense as present, past and present perfect and according to personal pronouns as first, second and third persons. In the previous method using POS corpus, inclusion of a verb is determined by the frequency of corpus. Therefore, it is not consistent whether the vocabulary also includes words of changed forms of the corresponding verb. The user would be confused if the present form (e.g. run) is recognized but the past form for the same word (ran) is not always recognized. In this section, in order to solve this problem, verb-related words are grouped according to their lemma and the method is proposed whether all of the root and changed forms of a verb in the same group are included. A lemma includes all changed forms of a verb according to tense, personal pronouns, singular and plural.

In the case of nouns, words are grouped as synonyms from Wordnet, so that the number of words in the same group is too big to include all of the words in the vocabulary. However, in the case of verbs, even if all of the root and changed forms are included in the lemma group, it involves no problem in generating vocabulary, since the number of words in each lemma group is small. Thus in order to generate the vocabulary on verbs, the verbs are grouped and analyzed according to lemma by statistic values counted from the British National Corpus (BNC) [19]. This is an analysis of vocabulary belonging to BNC, with statistics arranged with the frequency in BNC on lemma of root and changed forms of a word. Thus, the importance of verb is decided by the frequency for each lemma group. If the lemma is included in vocabulary, all of its changed forms are included altogether in the vocabulary.

3) Estimation of Relative Importance between Named Entities Using Knowledge base

Proper nouns have important information on spoken contents search, thus much more attention is required in vocabulary generation regarding proper nouns [20]. However, these proper nouns often become OOV (Out-Of-Vocabulary). According to [20], among the 65K word vocabulary, about 28% of words was proper nouns. As the list of person's names, we use the most common surnames, male and female names counted from the results of the 1990 Population Census of the U.S. [21]. For the named entities apart from person's names, the relative importance was decided through Google search on words in the list of named entities used in NYU OAK system [22].

IV. Category-based Small Footprint Language Model

In this chapter, the general structure of our proposed category-based LM is explained first. Vocabulary, denoted as V, is represented as $V=\{v_1, \ldots, v_j, \ldots, v_{|V|}\}$ where each v_j is a distinct word and $|V|$ is the total number of words in the vocabulary. Let w_i be i-th word in a sentence. Each w_i is an element of V. Define $f_v(w_i)$ as a function which returns the index of the word in V corresponding to w_i. A category is assigned to a group of words which have same syntactic functions or similar semantic meanings. The category set consisting of the whole categories, used in a category-based LM, is denoted as $C=\{c_1,\ldots, c_k, \ldots, c_{|C|}\}$ where each c_k is a distinct category and $|C|$ is the total number of categories. $f_c(v_j)$ is a mapping function between v_j and a set of possible categories related to v_j. $f_c(v_j)$ is a one-to-many function, because the syntactic or semantic functions of the same word are sometimes different. Hence, a word is possible to assign multiple categories. For example, syntactic functions of the word '*dream*' are sometimes noun and sometimes verb. Thus, f_c('*dream*')={'*noun*', '*verb*'}. Assuming that generation of a word depends only on its categories, we may write the word n-gram probability as :

$$P(w_i \mid w_{i-n+1},...,w_{i-1}) \approx \sum_{\forall c: c \in f_c(f_v(w_i))} P(w_i \mid c) \cdot P(c \mid w_{i-n+1},...,w_{i-1}) \quad (2)$$

where $(w_{i-n+1},\ldots,w_{i-1})$ is the word history of w_i. In the above equation, w_i is generated based on all of its corresponding categories and these categories are generated from the word history of w_i.

A category history is defined as a tuple of categories, which corresponds to phrases or word segments preceding to w_i. Define $f_{ch}(w_{i-n+1},\ldots,w_{i-1})$ to be a function as follows:

$$f_{ch}(w_{i-n+1},...,w_{i-1}) = \{(t_m,...,t_o,...,t_{i-1}) \mid t_o \in f_c(f_v(w_o))\} \quad (3)$$

where $i\text{-}n+1 \le m \le o \le i\text{-}1$. The number of elements in $(t_m, \ldots,t_o,\ldots,t_{i-1})$ depends on the length of phrases or word segments that match parts of the word history of w_i. The set of whole category histories used in a category-based LM, denoted as H, consists of elements of f_{ch} over all possible word histories. Define H as $H=\{h_1, \ldots, h_{|H|}\}$ where $|H|$ is the total number of category histories used in a category-based LM. The length of a category history is determined as the number of categories in the tuple of its corresponding category history. The maximum length among the lengths of all elements in H is denoted as l.

Assuming that the probability of observing a category in $f_c(f_v(w_i))$ depends on category histories related to w_i, the second term on the right-hand side of (2) may be decomposed as in (4):

$$P(c \mid w_{i-n+1},...,w_{i-1}) \approx \sum_{\forall h: h \in f_{ch}(w_{i-n+1},...,w_{i-1})} P(c \mid h) \cdot P(h \mid w_{i-n+1},...,w_{i-1}) \quad (4)$$

As described above, a category-based LM requires the estimation of three probabilities: a probability of a word for its corresponding category ($P(w_i|c)$), a probability of a category for its corresponding category history ($P(c|h)$), and a probability of a category history for its corresponding word history ($P(h|w_{i-n+1},\ldots,w_{i-1})$). As a result, the estimation of $P(w_i|c)$, $P(c|h)$, and $P(h|w_{i-n+1},\ldots,w_{i-1})$ are the major three components in the implementation of a category-based LM. Our proposed methods for memory requirement reduction are described in the following section.

In order to reduce the space complexity, our tree-based compact data structure is proposed in Section IV-A. In this tree, l is represented as the maximum level of leaf nodes. In Section IV-B, tree growing strategy at each node is described. Because the tree grows-up to cover all of category histories appeared in training corpus, it is necessary to devise a level-by-level tree growing strategy. In Section IV-C, our proposed independent assumption is introduced.

A. Tree-based Compact Data Structure for Estimating P(c|h)

It is likely that several category histories have the same starting category or category sub-histories. This property can be used to store category histories in a tree structure. In a tree structure, each node is associated with a particular category, so that paths originating at the root correspond to category histories. In this way each node represents a distinct history h, and associates with it a conditional probability distribution function $P(c|h)$. By not restricting the length of the individual paths in the tree, arbitrary length category histories are implemented. Nodes are labeled with the specific category history h which they represent, and the category which they are associated with. By applying a breadth first search, each node can be numbered according to the order of its traversal. This number can be used as an index for H. Any category history can be represented as a node in this tree, and each node can be mapped into a specific element in H. As a result, one-to-one mapping exists between the nodes in the tree and the elements in H. Therefore, estimating $P(c|h)$ is successfully implemented using the proposed tree.

B. Tree Growing Strategy

In general, a category history tree grows up to cover all of the category histories appearing in the training corpus. This causes two problems. First, the tree overfits to the training corpus. As a result, the best performance is not guaranteed for the test data. Second, the memory requirements for storing this tree are not well suited for the embedded environment.

In order to improve performance and reduce memory requirements, it is necessary to devise a level-by-level tree growing strategy. The simplest method is to restrict the maximum level of the tree, similar to a bigram or trigram word-based LM where the length of word history is limited to one or two, respectively. Another general strategy is pruning, which is based on a quality criterion [23]. This strategy is normally used in a variable-length category-based LM to look up longer lengths of histories to improve LM performance. It is based on the amount of improvement in LM performance, such as in terms of improvement in log likelihood, incurred by the inclusion of the node. At a specific node of the tree, a general approach of this tree growing strategy is as follows. First, collect all the category histories that have appeared in training corpus, which correspond to the node. Second, measure the improvement of LM performance using leaving-one-out cross validation and discard the node if it fails the criterion.

C. Decomposition of $P(h|w_{i-n+1},\ldots. w_{i-1})$ using Independent Assumption

The space complexity for the estimation of $P(h|w_{i-n+1},\ldots,w_{i-1})$ is measured as $O(|C|^l*|V|^{n-1})$. To reduce this intractable space complexity, we introduce an independent assumption that the generation of the category for the current word, w_i, is dependent only on w_i.

$P(h_g|w_{i-n+1},\ldots,w_{i-1})$ is re-estimated as in (5):

$$P(h_g \mid w_{i-n+1},...,w_{i-1}) \approx \prod_{o=m}^{o=i-1} P(c_o \mid w_o) \quad (5)$$

where $c_o \in f_c(f_v(w_o))$, $i\text{-}n+1 \le m \le o \le i\text{-}1$. Of course, the applied assumption is not mathematically correct. However, if the dependency is extended to cover only one previous word w_{i-1}, the resulting space complexity becomes intractable. Therefore, the applied assumption is reasonable in the embedded environment, although this assumption is not mathematically correct.

V. EXPERIMENTS

The evaluation metrics used in this paper are described in Section V-A. The details of experimental setups are explained in Section V-B. In Section V-C, the results are analyzed.

A. Evaluation Metrics

The effectiveness of a vocabulary generation method is evaluated in terms of the coverage of the vocabulary over the test domain.

The quality of an LM is assessed in terms of its overall memory requirement as well as its normalized perplexity. The perplexity [24] is based on concept of information theory as well as log probability.

Let v_j be j-th word in the vocabulary as defined in Chapter IV. As the current word w_i is one of words in the vocabulary, $\sum P(v_j \mid w_1,\ldots,w_{i-1})$ should be 1. However, due to decomposition assumption applied to a category-based LM, $\sum P(v_j \mid w_1,\ldots,w_{i-1})$ is not always 1. For a fair comparison between different LMs , it is necessary to normalize each $P(w_i|w_1,\ldots,w_{i-1})$ as $P'(w_i|w_1,\ldots,w_{i-1})$, which is equal to $P(w_i|w_1,\ldots,w_{i-1}) / \sum P(v_j \mid w_1,\ldots,w_{i-1})$.

The average value of the log normalized probability, denoted as *LNP*, can be determined as in equation (6):

$$LNP = \lim_{N\to\infty} -\frac{1}{N}\sum_{i=1}^{N}\log_2\left\{\frac{P(w_i \mid w_1,\ldots,w_{i-1})}{\sum_{j=1}^{|V|}P(v_j \mid w_1,\ldots,w_{i-1})}\right\} \quad (6)$$

This definition of *LNP* is closely related to the entropy of the LM. The normalized perplexity, denoted as *NPP*, can be determined as:

$$NPP = 2^{LNP} = \lim_{N\to\infty}\left\{P'(w_1,\ldots,w_i,\ldots,w_{|S|})^{-\frac{1}{|S|}}\right\} \quad (7)$$

where $|S|$ is the total number of words in a test sentence. Hence it can be viewed as the average branching factor of the LM. Therefore, the smaller the perplexity is, the better the LM. In this research, the normalized perplexity is used as an evaluation metric.

B. Experimental Setup

The test material was provided by LG Electronics Inc. The test reference was collected by transcribing the SMS messages of local mobile phone users in the United States. Two different corpora are collected by transcribing the SMS messages. The first corpus consists of 20K SMS sentences, where categories are tagged manually. These 20K sentences are denoted as SMS_MT_20K. The other 70K sentences which do not have any overlap with SMS_MT_20K, are collected using the same method. For this corpus, categories are tagged automatically, in accordance with the category that has the maximum frequency in SMS_MT_20K. The second corpus is denoted as SMS_AT_70K. TABLE VI illustrates the statistics of the two corpora.

TABLE VI. STATISTICS OF SMS CORPORA

Corpus	No. of Sentences	No. of Words
SMS_MT_20K	20,251	113,708
SMS_AT_70K	70,000	429,871

To prove the effectiveness of the domain independent vocabulary generation method, eight vocabularies were investigated as shown in TABLE VII. SMS is a text but is constituted with spoken style sentences, thus we generated a vocabulary based on word frequencies from American National Spoken corpus [25] as the baseline vocabulary (hereafter Voc_ANC_Spoken). We also generated the vocabulary according to the proposed method (hereafter Voc_KnBased). For each type of vocabulary, we set the size of vocabulary as 5K, 10K, 15K, and 20K.

TABLE VII. DESCRIPTION OF VOCABULARY

Vocabulary	Vocabulary Size	Vocabulary Generation Method
Voc_ANC_Spoken	5K, 10K, 15K, 20K	The frequency based method applied for American National Spoken corpus
Voc_KnBased	5K, 10K, 15K, 20K	The proposed method

In order to confirm the effectiveness of the proposed category-based small footprint LM (hereafter CBLM_SG), two different LMs were also implemented. A word-based trigram LM (hereafter WBLM) and a conventional category-based LM (hereafter CBLM_SRI) were implemented by CMU&Cambridge SLM Toolkit [26] and SRI Language Modeling Toolkit [27], respectively. CBLM_SRI has a constraint, which every word assigned to only one category. The descriptions of these LMs are shown in TABLE VIII. The quality of the LMs was evaluated in terms of normalized perplexity and memory requirement. In order to perform fair performance comparison, we restricted the maximum level of the category history tree as two, as the length of n-gram in WBLM can not exceed two, due to memory constraints.

TABLE VIII. DESCRIPTIONS OF LMS

LM	Description
WBLM	Word-based trigram LM implemented by CMU&Cambridge SLM toolkit
CBLM_SRI	Conventional category-based LM implemented by SRI language modeling toolkit
CBLM_SG	Proposed category-based small footprint LM

C. Results

The vocabulary coverages were measured on the test reference, SMS_MT_20K. TABLE IX shows the coverages of vocabularies according to vocabulary size. The coverages of Voc_KnBased were measured at 93.44%, 96.03%, 97.11% and 97.18% for 5K, 10K, 15K and 20K, respectively. The proposed method shows higher coverages for all cases (at least 13.68% relative improvement). In particular, when the size of vocabulary is 15K, the relative improvement reaches 33.41%. This proves that the proposed method is more effective than the conventional method of vocabulary generation for the domain with sparse data.

TABLE IX. COVERAGES OF VOC_ANC_SPOKEN AND VOC_KNBASED FOR SMSTEXT

Vocabulary Size	Coverage (%)		Relative Improvement
	Voc_ANC_Spoken	Voc_KnBased (proposed)	
5,000	92.40	93.44	13.68
10,000	94.71	96.03	24.95
15,000	95.66	97.11	33.41
20,000	96.44	97.18	20.78

TABLE X shows the results of quality comparison between WBLM, CBLM_SRI and CBLM_SG. Normalized perplexities were measured in accordance with 10 cross-validations. 90% of SMS_MT_20K is used as training data,

while the other 10% of SMS_MT_20K is used as test data.

The proposed category LM requires only 215KB which is 55% and 13% of the memory requirement for the CBLM_SRI and WBLM, respectively. It successively improves the performance, achieving 54.9% and 60.6% perplexity reduction compared to CBLM_SRI and WBLM in terms of normalized perplexity.

TABLE X. RESULTS OF QUALITY COMPARISON BETWEEN WBLM, CBLM_SRI AND CBLM_SG

	WBLM	CBLM_SRI	CBLM_SG (proposed)
Normalized Perplexity	127.30	111.25	50.13
Memory Requirement(KB)	1,652	387	215

TABLE XI shows the results of memory requirement comparisons as the size of training data increases. SMS_AT_70K is used as an additional training corpus. Sentences in SMS_AT_70K are put together in the training corpus. As a result, eight different training corpora are developed by increasing the number of sentences from SMS_AT_70K with the increment of 10K. The effects of the amount of training data are analyzed by changing that of training data. On the basis of the results shown in TABLE XI, we can conclude that the memory requirement ratio of CBLM_SG to WBLM (CBLM_SG/WBLM) varies around 0.13 in all the cases of the different numbers of sentences.

TABLE XI. RESULTS OF MEMORY REQUIREMENT OF WBLM AND CBLM_SG ACCORDING TO THE SIZE OF TRAINING DATA

No. of Sentences in Training Data	Memory Requirement(KB)		Memory Req. Ratio(B/A)
	WBLM(A)	CBLM_SG(B)	
18K	1,652	215	0.13
28K	1,814	246	0.14
38K	1,996	269	0.13
48K	2,119	290	0.14
58K	2,365	314	0.13
68K	2,774	354	0.13
78K	2,985	369	0.12
88K	3,160	380	0.12

VI. CONCLUSION

In this paper, domain independent vocabulary generation and its use in category-based small footprint LM in embedded environment were investigated to overcome two major constraints of conventional LM in embedded environment: memory capacity limitation and data sparsity. The proposed methods were evaluated using SMS texts. The proposed vocabulary generation method showed higher coverages, at least 13.68% relative improvement. The proposed category LM requires only 215KB which is 55% and 13% of the memory requirement for the CBLM_SRI and WBLM, respectively. It successively improves the performance, achieving 54.9% and 60.6% perplexity reduction compared to CBLM_SRI and WBLM in terms of normalized perplexity.

REFERENCES

[1] S. Young, "A Reivew of Large Vocabulary Continuous Speech Recognition," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 45-57, 1990.

[2] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK version 3.2)*, Cambridge University Engineering Department, 2002.

[3] K. Lee, H. Hon, and R. Reddy, "An Overview of the SPHINX Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 35-45, 1990.

[4] J. Novak, P. Dixon, and S. Furui, "An Empirical Comparison of the T^3, Juicer, HDecode and Sphinx3 Decoders," *Proc. Interspeech*, pp.1890-1893, 2010.

[5] M. Adda-Decker and L. Lamel, "The Use of Lexica in Automatic Speech Recognition," *Lexicon Development for Speech and Language Processing,* F. van Eynde, D. Gibbon (Eds.), Kluwer Academic, pp. 235-266, 2000.

[6] R. Rosenfeld, "Optimizing Lexical and N-gram Coverage via Judicious Use of Linguistic Data," *Proc. Eurospeech*, pp. 1763-1766, 1995.

[7] S. Adolphs and N. Shemitt, "Lexical Coverage of Spoken Discourse," *Applied Linguistics*, vol. 24, no. 4, pp. 425-438, 2003.

[8] P. Nation and R. Waring, "Vocabulary Size, Text Coverage and Word Lists," *Vocabulary: Description, Acquisition and Pedagogy*, N. Schmitt, M. McCarthy (Eds.), Cambridge University Press, pp. 6-19, 1997.

[9] S. Katz, "Estimation of Probabilities from Sparse Data for The Language Model Component of a Speech Recognizer," *IEEE Transaction on Acoustic, Speech and Signal Processing.*, vol. 35, no. 3, pp. 400-401, 1987.

[10] F. Bechet, Y. Esteve, and R. Mori, "Tree-based Language Model Dedicated to Natural Spoken Dialog System," *Proc. International Symposium on Computer Architecture*, pp. 207-210, 2001.

[11] C. Troncoso and T. Kwawahara, "Trigger-based Language Model Adaptation for Automatic Meeting Transcription," *Proc. Interspeech.*, pp. 1297-1300, 2005.

[12] I. Zitouni, K. Samili, and J. Haton, "Statistical Language Modeling Based on Variable-length Sequence," *Computer Speech and Language*, vol. 17, no. 1, pp. 27-41, 2003.

[13] I. Zitouni, "Backoff Hierarchical Class N-gram Language Models: Effectiveness to Model Unseen Events in Speech Recognition," *Computer Speech and Language.*, vol. 21, no. 1, pp.88-104, 2007.

[14] H. Yamamoto, S. Isogai, and Y. Sagisaka, "Multi-class Composite N-gram Language Model," *Speech Communication.*, vol. 41, no. 2, pp 369-379, 2003.

[15] P. Brown, V. Pietra, P. deSouza, J. Lai, and L. Mercer, "Class-based n-gram Models of Natural Language," *Computational Linguistics.*, vol. 18, no. 4, pp. 467-479, 1990.

[16] R. Kneser and H. Ney, "Improved Clustering Techniques for Class-based Statistical Language Modeling," *Proc. Eurospeech.*, pp.973-976, 1993.

[17] J. Stig, "Word Frequency and Text Type: Some Observations based on the LOB Corpus of British English texts," *Computers and the Humanities*, vol. 19, no. 1, pp. 23-36, 1985

[18] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[19] G. Leech, P. Rayson, and A. Wilson, *Word Frequencies in Written and Spoken English: Based on the British National Corpus*, Pearson ESL, 2001.

[20] R. Ordelman, A. Hessen, and F. Jong, "Lexicon Optimization for Dutch Speech Recognition in Spoken Document Retrieval," *Proc. Eurospeech*, pp. 1085-1088, 2001.

[21] J. Burger, J. Henderson, and W. Morgan, "Statistical Named Entity Recognizer Adaptation," *Proc. Natural Language Learning*, pp. 1-4, 2002.

[22] T. Hasegawa and S. Sekine, "Discovering Relations Among Named Entities from Large Corpora," *Proc. Association for Computational Linguistics*, pp. 415-442, 2004

[23] H. Blockeel, L. Raedt, and J. Ramon, "Top-down Induction of Clustering Trees," *Proc. International Conference on Machine Learning*, pp.55-63, 1998.

[24] P. Banerjee and H. Han, "Language Modeling Approaches to Information Retrieval," *Journal of Computing Science and Engineering*, vol. 3, no. 3, pp. 143-164, 2009.

[25] N. Schmitt and M. McCarthy, *Vocabulary: Description, Acquisition and Pedagogy*, Cambridge University Press, pp. 6-19, 1997.

[26] P. Clarkson and R. Rosenfeld, "Statistical Language Modeling Using the CMU-Cambridge Toolkit," *Proc. Eurospeech.*, pp. 2707-2710, 1997.

[27] A. Stolcke, "SRILM-an Extensible Language Modeling Toolkit," *Proc. International Conference on Spoken Language Processing*, pp. 901-904, 2002.