An Efficient Minimum Vocabulary Construction Algorithm for Language Modeling

Sina Lin[1], Zengchang Qin[1,3], Zehua Huang[1,2], and Tao Wan[4]

[1] Intelligent Computing and Machine Learning Lab
School of ASEE, Beihang University, Beijing, China
[2] School of Advanced Engineering, Beihang University, China
[3] Robotics Institute, Carnegie Mellon University, Pittsburgh, USA
[4] School of Medicine, Boston University, Boston, USA
zcqin@buaa.edu.cn

Abstract. In learning a new word by a dictionary, we first need to know a set of "basic words" which are frequently appeared in word definitions. It often happens that you cannot understand the word you looked up because there are still some words you do not understand in its definitions or explanations provided by the dictionary. You can keep looking up these new words recursively till they all can be well explained by some *basic words* you already knew. How to automatically find a minimum set of such basic words to define (or recursively define) the entire vocabulary in a given dictionary is what are going to discuss in this paper. We propose an efficient algorithm to construct the *Minimum Vocabulary* (MV) using the word frequency information. The minimum vocabulary can be used for language modeling and experimental results demonstrate the effectiveness of using the minimum vocabulary as features in text classification.

1 Introduction

The emergence of a complex language is one of the fundamental events of human evolution, some words are believed to be more complex than others because they present more precise semantic meanings that can be well explained by using some "basic words". In learning a new language, the dictionary is a powerful tool to learn a new complex word based on the words you already knew though some explanations are unavoidably reciprocal or circular, as "*hind, the female of the stag; stag, the male of the hind.*" When you are learning a new language, after you have grasped some basic words and start to use dictionary to learn yourself, it often happens that you cannot understand a word you just looked up because there are still some words you do not understand in its explanations provided by the dictionary. You can keep looking up these words till they all can be well explained by the basic words you knew. However, if you find yourself stuck in a recursive process of keeping looking up different new words or in a reciprocal process like the above *hind-stag* example, it means you may need a bigger set of "basic words". A simple question arose, *what are these basic words in a given dictionary?*

It's been a long standing question to find whether such a set of basic words exists that can be used to define the entire vocabulary in this fashion. This is called the *Minimum Vocabulary Problem* (MVP) [3]. MVP aims to find a minimum vocabulary set which

H. Jiang et al. (Eds.): IEA/AIE 2012, LNAI 7345, pp. 292–301, 2012.

can define (or recursively define) the entire vocabulary in one dictionary. MVP has been widely studied in literature, many of which focused on suggesting appropriate size of vocabularies for different level language learner [5,6]. These works are mainly based on manual analysis and corpus statistics, which are less informative in language modeling. For example, general service list [13] selected 2000 basic words for new English learners. Research for using an automatic algorithm to solve MVP could help us to uncover the intrinsic structure of human language by finding these basic "semantic bricks" for building a language. Fig. 1 gives a schematic illustration how a word "haphazard" can be explained by using other words in its definition from the Collins dictionary. This tree will be extended till all the leaf nodes are words in the minimum vocabulary (MV).

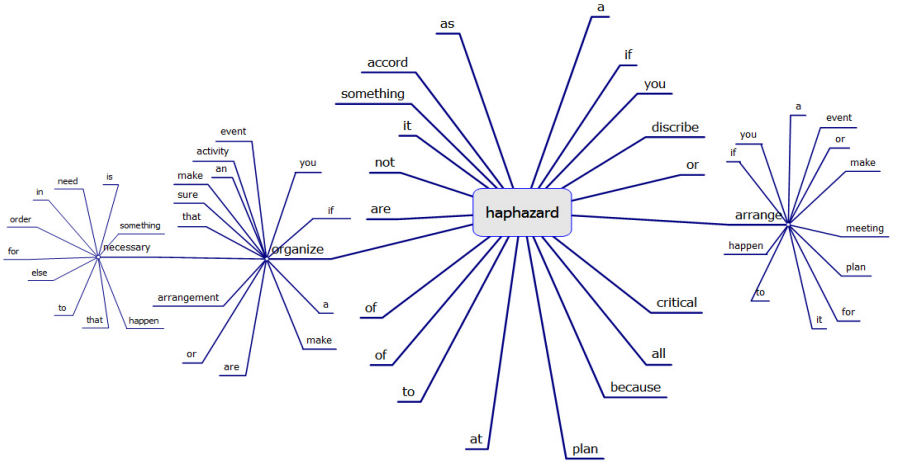

Fig. 1. An example of word dependency. All the leaf node words are belonging to the minimum vocabulary, they are used to explain a complex word "haphazard".

A computation solution of this problem is first proposed in [3], in which a graph model is used to reformulate the problem as calculating the basis of a definition graph and achieves fast computation using approximation. But this method requires much work on pre-processing and reduces the applicability. In this paper, we only focus on English language though the same methodology can be applied to many other alphabetic languages such as German, French and Spanish. We propose a new model by adopting the methodology of relational database design [11]. By transforming the whole dictionary into a functional space, we can model word-definition relations as a functional dependency of graphs. The problem is then simplified to seek the minimum closure of a set that shares the common strategy with calculating candidate key of a relational database [11]. However, if taking account of the enormous size of a dictionary, traditional methods would fail because of computational complexity. In this paper, we propose a fast approximation by combining each word with the prior knowledge of the word frequency and iteratively calculating the basic word set to construct the minimum vocabulary. Based on experimental studies, this method makes a good balance on both performance and computation efficiency.

This paper is organized as follows. In Section 2 we describe the formulation of the problem and a new algorithm is proposed. In Section 3, experimental results of three well known dictionaries are presented. We also introduce the concept of using the MV as features descriptors in language modeling and apply the model in text classification. Finally, the conclusions are given in Section 4.

2 Minimum Vocabulary Model

We use $\mathcal{D}$ to denote a given dictionary and $w \in \mathcal{D}$ is the word in $\mathcal{D}$. $\mathcal{E}(w)$ stands for the set of words in the definition (explanation) of w. We also define $w \in \mathcal{E}(w)$ for mathematical consistence because it is true that word can be explained by itself. The main idea is to set up a directional relation from w to its explanations $\mathcal{E}(w)$. Given the nature of this problem, mathematics of relational database can be used. For given two sets of words $S_i, S_j \subseteq \mathcal{D}$, we propose a dependency property developed from the similar concept in relational database.

Definition 1. *Given two sets of words S_i, $S_j \subseteq \mathcal{D}$, S_i depends on S_j, or S_j determines S_i (denoted by $S_j \Rightarrow S_i$) when:*

$$\forall w \in S_i : \mathcal{E}(w) \subseteq S_j$$

In other words, if we knew all the words in S_j, we can also know all the words in S_i because all the words in S_i can be explained by words in S_j. Any set S depends on itself based on the above definition, i.e.: $S \Rightarrow S$.

The following properties [1] hold for the dependency relation:

1. **Reflexivity:** If Y is a subset of X ($Y \subseteq X$), then $X \Rightarrow Y$.
2. **Augmentation:** If $X \Rightarrow Y$, then $X \cup Z \Rightarrow Y \cup Z$, for any word set Z.
3. **Transitivity:** If $X \Rightarrow Y$ and $Y \Rightarrow Z$, then $X \Rightarrow Z$.

These properties can be easily proved using the definition of relational dependency. Therefore, an unknown word set can be inferred from a set of basic words through definition relations, this can be well explained through the empirical experience that a new (complex) word could be learnt from a certain amount of very simple words.

Definition 2. *The closure of S, or S^+, is the set of words that can be determined by S, or $S \Rightarrow S^+$ where:*

$$S^+ = \{x | w \in S, x \in \mathcal{E}(w)\}$$

The solution of MVP is about to find a minimum set of S that $S^+ = \mathcal{D}$.

2.1 MV Construction Algorithm

Previous research in relational database design provides a few efficient solutions for calculating the minimum set of all relational attributes which is referred as *candidate key*. Saiedian and Spencer [12] proposed a graph method to extract candidate key with

Algorithm 1. Minimum Vocabulary Construction Outline

```
1: procedure MINIMUM VOCABULARY(𝒟)
2:     S = 𝒟
3:     repeat
4:         for all w_r ∈ S do
5:             if w_r ∈ {S − ∑_r w_r}^+ then
6:                 S ← {S − w_r}
7:             end if
8:         end for
9:     until No more word w_r can be removed from S
10:    return S
11: end procedure
```

time complexity of $O(kn^2)$, where n is the number of items and k is the number of dependencies between these items. Since they focused only on database design, when the size of database is small, the problem is tractable. However, for our problem, a fair dictionary commonly has over 30000 words and millions of dependency relations, their method is computationally inefficient.

In out approach, the main idea for MV construction is simple. We start from a set S assigned with $\mathcal{D}$ ($S = \mathcal{D}$), then we iteratively remove redundancy word w_r that can be explained (or recursively explained) by the rest of words denoted by the set $\{S - \sum_r w_r\}$. The removing word satisfies that $w_r \in \{S - \sum_r w_r\}^+$. The algorithm terminates when no more word can be removed from S to satisfy the above conditions. The pseudo-code is shown in Algorithm 1. However, it hasn't consider the key factors such as the removing order and the closure computation, both of which are curial for fast computation and effective performance, that will be discussed in the next section.

2.2 Familiarity and Frequency of Words

As we can see from the previous section, the key problem is to decide a preference of removing order of the words. Based on empirical knowledge of language, people tend to use words they are more familiar with to explain those are not, we therefore need to choose the attribute implying "familiarity" to evaluate word preference. There is a rich literature in human perception of "familiarity" and some insightful discussions on this topic are available in [2,8]. Here we adopt the simplest familiarity measure in terms of word frequency.

In this research, frequency statistics on BNC database [7] is used to assign each word with an attribute of frequency. This attribute defines the preference of removing redundant word in applying Algorithm 1. In details, we sort the removing words sequentially based on word frequency in ascending order. The lower frequency a word has, the higher possibility to be removed from the basic word set. $R[w]$ is used denote the frequency of word w.

The closure computation for a large set has a high computational complexity. Therefore, in order to propose a reliable solution for the MVP, we propose an efficient algorithm for closure computation. Line 5 in Algorithm 1 of closure computation can be

modified from calculating the entire set closure to measuring whether a word can be explained by a set of high frequency words. We present details of the proposed method on calculating in Algorithm 2.

For each pair $(w, R[w])$, we calculate the max preference word in its definition word set $\mathcal{E}(w)$. We define the word with maximum frequency in $\mathcal{E}(w)$ is

$$mf(w) = \max_{i \in \mathcal{E}(w)} R[i] \tag{1}$$

If $mf(w) < R[w]$, it means w can be explained by words with higher preference. Then $R[w]$ can be replaced with $mf(w)$. If not, it means w temporarily can not be removed from the word set, then w becomes the candidate of basic word. For each iteration of scanning overall words, we get all candidate basic words $\mathcal{S}$. Since all words can be explained by $\mathcal{S}$, we have $\mathcal{S}^+ = \mathcal{D}$. Note preferences of all words are updated throughout each iteration, some words might change their preferences. We need reprocess all words until the full coverage of all words. The algorithm runs in $O(kn)$, where n is the size of dictionary and k is the number of process iterations. The pseudo-code is given in Algorithm 2.

Algorithm 2. Word Frequency Based Minimum Vocabulary Construction

```
    procedure MINIMUM VOCABULARY(D)
 2:     Sort w ∈ D based on word frequency R[w]
        for w ∈ D do
 4:         RemovingTag[w] ← False
        end for
 6:     repeat
            for all w ∈ D do
 8:             if mf(w) < R[w] then
                    R[w] ← mf(w)
10:                 RemovingTag[w] ← True
                end if
12:         end for
        until convergence
14:     for all w ∈ D do
            if RemovingTag[w] == False then
16:             add w to S
            end if
18:     end for
        return S
20: end procedure
```

3 Experimental Studies

Since the model is heavily based on the word dependency relationship. For the same word, its definition may not be identical in different dictionaries. In our experiments, we tested 3 well known dictionaries using the new proposed algorithm: *Collins,*

Webster and *Longman* [1]. In these dictionaries, each phrase is consisted of phonetic symbols, several definitions and example sentences. In the following experiments we only consider the word itself and the first of its definitions, other semantic information is ignored at this stage . We lemmatize each word using Python NLP Toolkit [9]. Word frequency introduced in Section 2.2 is computed throughout word frequency list [2] based on BNC Corpus [7]. We perform the same lemmatization and assign each word in dictionary with a frequency $R[w]$.

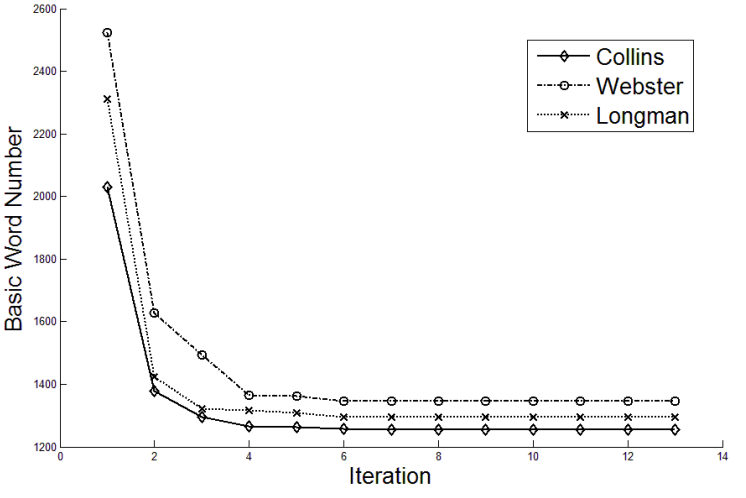

Fig. 2. Minimum vocabulary for Collins, Longmand and Merrian-Webster based on the frequency-based minimum vocabulary construction algorithm

3.1 Minimum Vocabulary of Dictionaries

Each dictionary has average 30000 words, Algorithm takes about 1 minute to finish the computation. The sizes of MV for Collins, Longman and Merrian-Webster dictionary are 1256, 1295 and 1346, respectively. Fig. 2 illustrates the number of words in the MV with the increasing number of iterations. The results show that the size of MV converges very fast and becomes stable after about 5 iterations. For the 3 given dictionaries, the sizes of MV are very similar.

Table 1 shows the results of overlapping of MV in 3 dictionaries. The common basic words appear in all dictionaries are 672, taking up about 50% in each dictionary's basic word set. This can be explained by that dictionary use simple and high preference (frequency) words in word definitions. Therefore, most words appear in definition would be among the small set of high frequency words and the MVs for different dictionaries have a high overlapping.

[1] The source of these dictionaries can be obtained for free from the following links: `http://debian.ustc.edu.cn/debian-uo/dists/sid/ustc/pool/stardict/`

[2] Available at the link: `http://www.kilgarriff.co.uk/bnc-readme.html`

Table 1. The number of common words in the MVs and the percentages of them in the three given dictionaries: Collins, Longman and Webster

Combination	Number	Percentage of Common Words in Dictionary
Collins ∩ Longman	824	Collins: 66% Longman: 64%
Collins ∩ Webster	861	Collins: 69% Webster: 64%
Longman ∩ Webster	957	Longman: 74% Webster: 72%
Collins ∩ Longman ∩ Webster	672	Collins:54% Longman: 51% Webster: 50%

3.2 Minimum Vocabulary Properties

In the framework of using MV for language analysis. A word may be explained in several layers like a tree, where all the leaf nodes are the words in the MV (e.g., see Fig. 1). The maximum level of the MV interpretation of a particular word may have some implications on their semantic complexities. More layers a word has, more semantically difficult this word is. We summarize the statistics of word levels across the whole dictionary. The left-hand figure of Fig. 3 shows the histogram of the word levels. The right-hand figure illustrates the accumulated percentage of words under the given word level. For example, 80% of words are under the level 10 and nearly 90% of words are under the level 15.

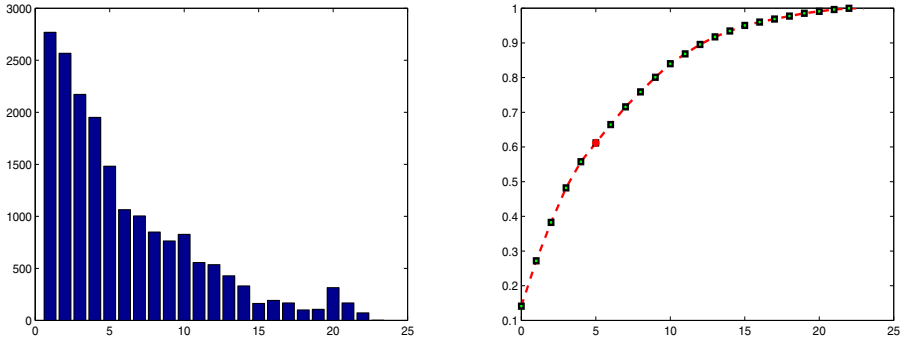

Fig. 3. Left-hand: histogram of word levels. Right-hand: accumulative percentage of words under the given word level.

In order to validate effectiveness of using the MV as the language model, we analyze the similarities between MVs of synonym pairs comparing to the similarities between non-synonym pairs. The similarity measure between to MVs can be defined by:

Definition 3. *Given two sets of words W_1 and W_2 , the relative similarity degree is defined by the ratio between the intersection of W_1 and W_2 and the union of these two sets.*

$$sim(W_1, W_2) = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|} \tag{2}$$

where $|\cdot|$ represents the cardinality of a set.

We collect a set of synonym pairs from the Oxford Thesaurus [3]. Phrases and stop words are removed. Considering that our focus are complex words, we also removed the synonyms in the given MV. In our experiment, we take two pairs of synonyms where A and B is one pair and C and D is another pair, i.e.:

$$A \leftrightarrow B,\ C \leftrightarrow D \tag{3}$$

We then calculate similarities between each pair of words, the following relations should hold:

$$sim(A, B) \geq sim(A, C)$$

$$sim(C, D) \geq sim(B, C)$$

We calculated 1650 pairs of synonyms and obtained 75.2% of which satisfy the above relations. This experiment can verify that MV based measure can reflect certain semantic relations with good confidence. In the next section, we will use MV as language features in text classification.

3.3 Document Feature Descriptor

Language modeling is an important topic in computational linguistics, techniques such as latent semantics indexing, topic models were used to map a text into a low dimension semantic space [14]. In such a space, different natural language problems can be studied by capturing the semantic meaning of the original text, e.g., question answering [10]. In this study, the MV also can provide the semantic relations between complex words and a text can be modeled in the MV space, that is how the MV can be used as descriptors for natural language modeling.

The MV with approximately 1300 words are obtained based on proposed algorithm. Words in the MV form the basic structure of a language. The meaning of the MV is highly compressed and may help to uncover intrinsic relations between words. For example, some cognates and synonyms may derive from same basic word ancestors. This property offers basic words a potential usage for a sound document descriptor. To demonstrate this, we employ the MV model to construct a document feature descriptor and apply it to text mining.

We denote $\mathcal{S}(w)$ as the basic words that w depends on, and $F(w_i)$ is the frequency of the ith basic word in $\mathcal{S}(w)$. The major purpose of this feature descriptor is to represent document using basic word histogram. To apply this, we first replace all the words by using basic words and calculate the basic histogram H_M.

We evaluate the MV descriptor on TechTC-100 Test Collection [4]. We use preprocessed feature vectors provided from this dataset, in which texts were simply tokenized and digitalized, no further processing was employed such as TF-IDF and lemmatization. We retrieve the feature descriptor by computing H_M based on the Minimum Vocabulary set of Collins Dictionary and obtain a 1253 dimension vector to represent each document. We compare our descriptor with original feature vector provided from the dataset. The classification tasks is performed between two classes of documents. We

[3] `http://thesaurus.com/browse/Oxford`

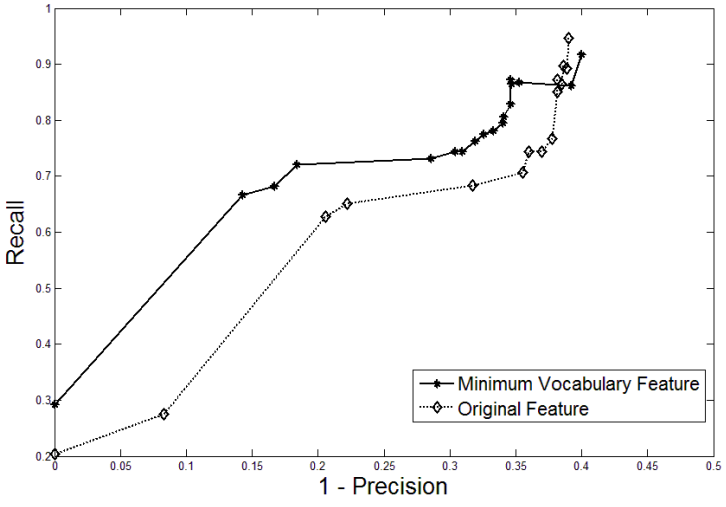

Fig. 4. ROC curve of classification by using MVs as text features

select linear SVM as the classifier. The recall and precision results is shown in Fig. 4. As we can see from the figure, the MV descriptor outperforms the original word feature significantly. The time consumption is also benefited from the downsize of feature vector, it reduces from 3.745s to 1.042s on a Dual Core 500MHZ machine.

4 Conclusions

This paper proposed an efficient computation method for the Minimum Vocabulary Problem. We proposed a new algorithm to construct the MV for a dictionary in order to investigate the word-explanation relationship by employing the word frequency regularizer. The empirical studies on three well known dictionaries are given. We also studied the properties of MV and use it in language modeling. The MV can be considered as the most basic "semantic bricks" for a language. Some initial investigations with experimental results of using MV features in text classification are given. Future works will study how to use the MV to solve other natural language processing problems.

Acknowledgment. This work is partially funded by the NCET Program of MOE, China and the SRF for ROCS. The second author also thanks the China Scholar Council for visiting fellowship (No. 2010307502) to CMU.

References

1. Armstrong, W.W.: Dependency structures of data base relationships. Information Processing 74, 580–583 (1974)
2. Cancho, R.F.I., Solé, R.V.: Least effort and the origins of scaling in human language. Proceedings of the National Academy of Science 100, 788–791 (2003)

3. Chandrasekharan, N., Sridhar, R., Iyengar, S.: On the minimum vocabulary problem. Journal of the American Society for Information Science 38(4), 234–238 (1987)
4. Gabrilovich, E., Markovitch: Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In: ICML (2004)
5. Hazenberg, S., Hulstun, J.H.: Defining a minimal receptive second-language vocabulary for non- native university students: An empirical investigation. Applied Linguistics 17(2), 145–163 (1996)
6. Hirsh, D., Nation, P.: What vocabulary size is needed to read unsimplified texts for pleasure? Reading in a Foreign Language 8, 689–696 (1992)
7. Kilgarriff, A.: Putting frequencies in the dictionary. International Journal of Lexicography 10(2), 135–155 (1997)
8. Kilgarriff, A.: Using word frequency lists to measure corpus homogeneity and similarity between corpora. In: ACL-SIGDAT Workshop on Very Large Corpora, pp. 231-245 (1997)
9. Loper, E., Bird, S.: NLTK: The natural language toolkit. In: Proceedings of the ACL 2002 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, vol. 1, pp. 63–70 (2002)
10. Qin, Z., Thint, M., Huang, Z.: Ranking Answers by Hierarchical Topic Models. In: Chien, B.-C., Hong, T.-P., Chen, S.-M., Ali, M. (eds.) IEA/AIE 2009. LNCS (LNAI), vol. 5579, pp. 103–112. Springer, Heidelberg (2009)
11. Ramakrishnan, R., Gehrke, J.: Database Management Systems. McGraw-Hill, Inc. (1999)
12. Saiedian, H., Spencer, T.: An efficient algorithm to compute the candidate keys of a relational database schema. The Computer Journal 39(2), 124–132 (1996)
13. West, M.P.: A General Service List of English Words. Longman (1976)
14. Zhao, Q., Qin, Z., Wan, T.: What Is the Basic Semantic Unit of Chinese Language? A Computational Approach Based on Topic Models. In: Kanazawa, M., Kornai, A., Kracht, M., Seki, H. (eds.) MOL 12. LNCS (LNAI), vol. 6878, pp. 143–157. Springer, Heidelberg (2011)