論文の内容の要旨

論文題目　Large Scale Machine Learning for Practical Natural Language Processing
（大規模機械学習による現実的な自然言語処理）

氏名　岡野原　大輔

I present several efficient scalable frameworks for large scale natural language processing (NLP). Corpus-oriented NLP has succeeded in a wide range of tasks like machine translation, information extraction, syntactic parsing and information retrieval. As very large corpora are becoming available, NLP systems should offer not only high performance, but also efficiency and scalability. To achieve these goals, I propose to combine online learning algorithms, string algorithms, data structures and sparse parameter learning.

The difficulties in large scale NLP can be decomposed into the following: (1) massive amount of training examples, (2) massive amount of candidate features and (3) massive amount of candidate outputs. Since solutions for (1) have already been proposed (e.g. online learning algorithms), I will focus on (2) and (3).

For problem (2), I consider document classification with ``all substring features''.

Although all substring features would be effective for determining the label of a document, the number of candidate substrings is quadratic to the length of a document. Therefore a naive optimization with all substrings requires prohibitively large computational cost. I show that statistics of substring features (e.g. frequency) can be summarized into a few classes much smaller than a document length. Moreover, by using enhanced suffix arrays, these effective features can be found exhaustively in linear time without enumerating all substring features. The experimental results show that the proposed algorithm achieved the higher accuracies than state-of-the-art methods. Moreover the results also show the scalability of our algorithm; effective substrings can be enumerated from one million documents in 20 minutes.

Another important part of the solution to case of (2) is `` combination features". In NLP a combination of original features could be most effective for the classification. Although candidate combination features are exponentially many, effective ones are very few. I present a method that can effectively find all such effective combination features. This method relies on a grafting algorithm, which incrementally adds features from the most effective one. This looks a greedy algorithm, but it can converge to the global optimum. To find such effective features, I propose a space efficient online algorithm to calculate the statistics of combination features with a simple filtering method. Experimental results show that the proposed algorithm achieve comparable or better results than those from other methods, and its result is very compact and easy to interpret.

For problem (3), I consider language modeling, which can be used to predict the next word given previous words as context or to discriminate correct sentences form incorrect ones. Since the candidate words are very many, only simple generative models (e.g. N-gram) are used in practice. I propose two novel language models that support efficient training and inference, and can capture more complex features. The first one is a Discriminative Language Model (DLM) that directly classifies a sentence as correct or incorrect. Since DLM need not to define a generative model, it can use any type of features such as the existence of a verb in the sentence. To

obtain negative examples for training, I propose to use pseudo-negative examples, sampled from generative language model. I found that that DLM achieved 75% accuracy in the task of that discriminating positive and negative sentences, though N-gram model or the parsing cannot discriminate these at all.

The second language model is a Hierarchical Exponential Model (HEM). In HEM, we build a hierarchical tree where each candidate output corresponds to a leaf, and a binary logistic regression model is associated with each internal node. Then, the probability of an output is given by the product of the probabilities of each internal node in the path from the root to the corresponding leaf. While HEM can use any type of features, it supports efficient inference. Moreover it supports the operation that finds the most probable output efficiently, which is fundamental in efficient LMs. I conducted experiments using HEM and other language models, and show that this model can achieve the higher performance than others.