Master's Thesis

An Empirical Study on Reduction of Parameter Redundancy in a Weight Matrix in a Biaffine Parser

Tomoki Matsuno

March 15, 2019

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Master of ENGINEERING

Tomoki Matsuno

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Satoshi Nakamura	(Co-supervisor)
Associate Professor Masashi Shimbo	(Co-supervisor)
Assistant Professor Hiroyuki Shindo	(Co-supervisor)

An Empirical Study on Reduction of Parameter Redundancy in a Weight Matrix in a Biaffine Parser*

Tomoki Matsuno

Abstract

Currently, the biaffine classifier has been attracting attention as a method to introduce an attention mechanism into the modeling of binary relations. For instance, in the field of dependency parsing, the Deep Biaffine Parser by [4] has achieved state-of-the-art performance as a graph-based dependency parser on the English Penn Treebank and the CoNLL 2017 shared task. In this paper, we show that the number of parameters in biaffine classifiers, $O(n^2)$ (n is the number of dimensions), is actually redundant by showing that reduction of the number of parameters improves the parsing performance. We reduce the number of parameters by assuming either symmetry or circularity of weight matrices. In our experiments on the CoNLL 2017 shared task dataset, our model achieved better or comparable accuracy on most of the treebanks with more than 16% parameter reduction.

Keywords:

natural language processing, dependency parsing, biaffine transformation, diagonalization

*Master's Thesis
Nara Institute of Science and Technology, March 15, 2019.

双アフィン構文解析器の重み行列における パラメータ削減の実証研究*

松野智紀

内容便覧

近年，双アフィン分類器は注意機構を二項関係のモデル化に導入するための手法として注目を集めている．例えば，係り受け解析では， Dozat らによる Deep Biaffine Parser が English Penn Treebank および CoNLL 2017 shared task においてグラフ型係り受け解析器としての最高精度を達成した．本稿では，双アフィン分類器中のパラメータ数 $O(n^2)$ （n は次元数）が冗長であることをパラメータ数の削減によって性能が向上するという結果によって示す．我々は重み行列に対称行列および巡回行列であるという仮定を置くことによってパラメータ数を削減する．我々の CoNLL 2017 shared task データセットにおける実験では，我々のモデルは，16%以上のパラメータを削減しながら，ほとんどのツリーバンクにおいて比較手法と比べ同等かそれ以上の精度を達成した．

キーワード:

自然言語処理，係り受け解析，双アフィン変換，対角化

*奈良先端科学技術大学院大学情報科学研究科情報処理学専攻修士論文 2019年3月15日.

Acknowledgements

本論文の執筆にあたり多くの人にご助力をいただきました。ここで皆様に深く感謝申し上げます。

松本裕治教授をはじめとする自然言語処理学研究室のスタッフの皆様には、様々な助言・指導をしていただきました。また、研究に集中できる最高の環境を提供していただきました。

知能コミュニケーション学研究室の中村哲教授には、ゼミナール等で有益なアドバイスをいただきました。

産業技術総合研究所の能地宏先生には、私にとって初めての学会発表となった研究テーマにおいて、研究の進め方、論文執筆、ポスター制作に至るまで、手厚い指導をしていただきました。

大阪大学の林克彦助教には、本修士論文の元となるアイディアをいただき、初めての国際会議投稿まで多岐にわたるご指導を賜りました。また、インターンシップ先での研究にも技術顧問として参加していただき、有益なコメントを多数いただきました。

自然言語処理学研究室の同期や先輩、後輩の皆様とは、研究についての議論を活発に行い、切磋琢磨することで、研究能力の底上げをすることができました。また、多様性豊かな面々と過ごすことで、人生をより味わい深く感じられるようになりました。

最後に、私を経済的にも精神的にも支えてくれた家族に深く感謝申し上げます。

Contents

List of Figures

1 Introduction

Recently, methods based on attention mechanisms have been used in various fields of natural language processing [1, 17]. In tasks which deal with **binary relations** that take two words as arguments, such as dependency parsing, a good number of these models have achieved high performance [10, 8, 4].

Biaffine transformation is a method to incorporate an attention mechanism into binary relations, which was proposed by [4] (following them, we call this method a **biaffine classifier**). They achieved the state-of-the-art performance among graph-based dependency parsers for the English Penn Treebank. In addition, the state-of-the-art transition-based parser on the English Penn Treebank uses a biaffine classifier to evaluate the probability distribution of a word coming into the stack at each parsing step [12].

While biaffine transformation has rich expressiveness for modeling binary relations, the weight matrix (bilinear term) contains $O(n^2)$ parameters (where n is the number of dimensions). This large number of parameters can result in a high degree of freedom for the model, which might cause overfitting, especially when a large number of training samples are not available [15].

In this thesis, we attempt to reduce the number of parameters by introducing either a symmetry or circularity assumption to the weight matrix of a biaffine classifier. Under either assumption, we can vectorize the matrix and reduce the space complexity to $O(n)$. Additionally, the time complexity becomes $O(n)$ in the case of symmetry and $O(n \log n)$ in the case of circularity with the fast Fourier transform. Furthermore, while the expressiveness of the model is restricted under the symmetry assumption*, a model under the circularity assumption is able to express asymmetrical relations.

*Denote the score of the bilinear term by $f(a, b)$ when a word pair a, b has a dependency relation. In this case, if the bilinear term is symmetrical, $f(a, b) = f(b, a)$. Such a scoring function is not appropriate for the expression of a directed edge.

In our experiments, we imposed constraints on the biaffine classifiers of the deep biaffine parser and examined their effects on the accuracy of the model. For our experiments, we used the dependency parsing dataset from the CoNLL 2017 shared task [22]. We chose four languages which have a relatively large amount of training examples and four languages which have fewer. From our experiments, we found that applying the circularity assumption led to the model outperforming the baseline for most of the languages.

2 Background

2.1 Dependency Parsing

Dependency Parsing is a field in natural language processing which analyzes the structures of sentences, and which originated from the traditional linguistic theory of dependency grammar. This field has attracted attention in many applications such as machine translation and information extraction since it can explicitly incorporate the syntactic structure of sentences into these tasks. Compared to phrase structure grammar, dependency structure is a more appropriate approach for languages with relatively flexible word order such as Japanese.

In dependency grammar, we assume that the syntactic structures consist of binary, asymmetric relations between words, which are called **dependency relations**. A pair of words have a dependency relation if one of them syntactically governs the other. Here, the word and its governed word are called the **head** and the **dependent**. Figure 2.1 shows an example of an English sentence which is annotated with dependency relations. Every arc has a label which represents the type of the syntactic relation between the dependent and the head. For example, the word "he" in the sentence is dependent on the word "had" and the label "nsubj" on the arc shows that the dependent is the nominal subject in relation to the head. It is worth noting that there is a dummy token "root" inserted in the sentence. This is for simplification of the formal definition and implementation. Since every word has a head, it is more straightforward to say that the head of the word "had" is the "root" instead of saying that the word had has no head.

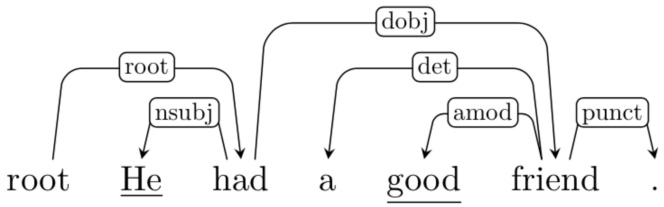

Figure 2.1: An example of a dependency tree.

2.2 Universal Dependencies

Universal Dependencies is a project which aims to develop a cross-linguistically consistent syntactic annotation scheme for various languages. The annotation scheme is based on (universal) Stanford dependencies [3], Google universal part-of-speech tags [16], and the Interset interlingua for morphosyntactic tagsets [20]. The Conference on Computational Natural Language Learning (CoNLL) has held a series of shared tasks on Multilingual Parsing from Raw Text to Universal Dependencies and has provided treebanks [22, 21] annotated with the universal dependencies annotation scheme. The use of muliti-lingual treebanks annotated consistently enables the analysis of the effectiveness of a method in a language-independent manner. The experiments in this thesis are conducted on several treebanks from CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies [22].

2.3 Graph-based Dependency Parser

To build a parser which derives the most probable dependency tree for a given sentence, one way is to make use of a traditional algorithm for directed graphs developed in the field of **Graph Theory**. The **graph-based dependency parser** first scores the edges on a given complete graph where the nodes are the words in a given sentence and the edges are the relations between each pair of the words. The score function is defined in equation 2.1, where $score(G)$ is a score for a given graph $G = (V, A)$ with V vertices and A arcs.

$$score(G) = score(V, A) \tag{2.1}$$

The models used in our experiments are called **arc-factored models**, which learn the score function h (equation 2.2) for the score $\lambda_{(w_i,r,w_j)}$ for a given arc consisting of two nodes: a head candidate w_i and a dependent candidate w_j, as well as a relation between them r.

$$\lambda_{(w_i,r,w_j)} = h(w_i, r, w_j) \tag{2.2}$$

Having defined the score function for an edge, we can subsequently define one for a complete directed graph by scoring every edge in the graph.

$$\lambda_{(w_i,r,w_j)} \in \mathbb{R} \; for \; each \; (w_i, r, w_j) \in A \tag{2.3}$$

The model then obtains the most probable dependency tree by running the Maximum Spanning Tree (MST) algorithm on the complete graph whose edges are scored.

3 Deep Biaffine Parser

Models introduced in this thesis are based on the **Deep Biaffine Parser** proposed by [4]. They achieved the state-of-the-art accuracy on the CoNLL 2017 shared task for Universal Dependencies [22].

This model receives a sequence of word and POS tag pairs and calculates the probability of an arc between each pair of words, as well as a syntactic function label for each arc. For calculation of scores, it uses Long Short-Term Memories (LSTMs), Multi-Layer Perceptrons (MLPs) and biaffine classifiers.

In the following sections, first, we explain the biaffine transformation which is the essential part of a biaffine classifier. We omit explanations for the LSTM and MLP for the sake of simplicity. Then, we give an overview of the model. It is worth noting that the structure of the model is different from that in [4], in that it utilizes character level information. This is because we used an updated version of the model that was made for the shared task [5].

3.1 Biaffine Transformation

For the dependency parsing score functions, we use the biaffine transformation shown below to model binary relations. Here, $\oplus$ stands for concatenation of vectors. The first term on the right side represents relatedness score, and the second term the score of v_i and v_j appearing independently. b is a bias term.

$$g(\mathbf{v}_i, \mathbf{v}_j) = \mathbf{v}_i^{\mathrm{T}} \mathbf{A} \mathbf{v}_j + (\mathbf{v}_i \oplus \mathbf{v}_j)^{\mathrm{T}} \mathbf{b} + b \tag{3.1}$$

3.2 Structure of the Model

The structure of this model is as follows:

1. First, this model takes a sequence of word and POS-tag pairs. It uses a unidirectional LSTM to encode each word's character-level information into a vector. It then sums this vector with a separate token-level word embedding.

2. The model then concatenates the vectors obtained above with POS embeddings and encodes them via a three layer bidirectional LSTM. $\mathbf{y}_i$ denotes a vector made by concatenating the hidden states (not including cell states) of the forward and backward LSTMs, which corresponds to the i th word w_i.

3. These outputs are then transformed with MLPs. Here, we use separate MLPs for dependents and heads in the prediction of arcs and labels. $\mathbf{v}_i$ and $\mathbf{v}_j$ represent the dense vector representations for the ith word and the jth word in the sentence, which in this case are the head candidate and the dependent candidate, respectively.

$$\begin{aligned} \mathbf{v}_i^{arc-head} &= \mathrm{MLP}^{arc-head}(\mathbf{y}_i), \\ \mathbf{v}_j^{arc-dep} &= \mathrm{MLP}^{arc-dep}(\mathbf{y}_j), \\ \mathbf{v}_i^{label-head} &= \mathrm{MLP}^{label-head}(\mathbf{y}_i), \\ \mathbf{v}_j^{label-dep} &= \mathrm{MLP}^{label-dep}(\mathbf{y}_j). \end{aligned} \tag{3.2}$$

   where $\mathbf{v}_i^{arc-head}, \mathbf{v}_j^{arc-dep} \in \mathbb{R}^n$
   and $\mathbf{v}_i^{label-head}, \mathbf{v}_j^{label-dep} \in \mathbb{R}^m$

4. The scores of arcs between each word pair are calculated using a biaffine transformation.

$$s_{i,j}^{(arc)} = \mathbf{v}_i^{arc-head\mathrm{T}} \mathbf{W} \mathbf{v}_j^{arc-dep} + \mathbf{v}_i^{arc-head\mathrm{T}} \mathbf{b}^{(arc)}. \tag{3.3}$$

5. Running Chu-Liu/Edmonds algorithm [2, 6] on the scores calculated in 3.3, we obtain a spanning tree that maximizes the total score.

6. The model then computes the score $s_{i,j}^{(l)}$ of assigning a label l ($l \in \{1, 2, ..., L\}$; L: the number of labels) to the arc between the head word w_i and its predicted dependent word w_j. The equation is defined as follows:

$$
\begin{aligned}
s_{i,j}^{(l)} &= \mathbf{v}_i^{label-head\,\mathrm{T}} \mathbf{U}^{[l]} \mathbf{v}_j^{label-dep} \\
&+ \left(\mathbf{v}_i^{label-head} \oplus \mathbf{v}_j^{label-dep}\right)^{\mathrm{T}} \mathbf{b}^{[l]} \\
&+ b^{[l]}.
\end{aligned} \tag{3.4}
$$

Here, a separate weight matrix $\mathbf{U}^{[l]}$, weight vector $\mathbf{b}^{[l]}$ and bias $b^{[l]}$ are used for each label. The first term on the right side of the equation 3.4 represents the score of assigning the label l to the arc between dependent w_i and head w_j. The second term represents the score of the label when the dependent and head are given independently.

In our experiments, $\mathbf{W} \in \mathbb{R}^{n \times n}$ and $\mathbf{U}^{[l]} \in \mathbb{R}^{m \times m}$ account for about 17% of the parameters in the deep biaffine parser. By reducing the number of these parameters, we can expect not only improved memory efficiency, but also less overfitting.

4 Proposed Method

In this section, we introduce our two proposed methods to reduce the number of parameters of the model. We impose either a symmetry or circularity constraint on the weight matrices $\mathbf{W}$ of (3.3) and $\mathbf{U}^{[l]}$ ($\forall l \in \{1, 2, \ldots, L\}$) of (3.4).

4.1 Symmetric Matrix Constraint

This method assumes that the weight matrices of the bilinear terms are **symmetric matrices** and are thus diagonalizable. As a result, we can transform the bilinear term of the score functions into a "triple inner product" of two input vectors and a weight vector.

Diagonalization of the Weight Matrix in the Bilinear Term

When a matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ is symmetric, it can be diagonalized by an orthogonal matrix $\mathbf{O} \in \mathbb{R}^{n \times n}$ as below:

$$\mathbf{W} = \mathbf{O}\mathrm{diag}(\mathbf{w})\mathbf{O}^{\mathrm{T}}$$

where $\mathbf{w} \in \mathbb{R}^n$ consists of the eigenvalues of $\mathbf{W}$ and $\mathrm{diag}(\mathbf{w}) \in \mathbb{R}^{n \times n}$ represents the diagonal matrix whose diagonal elements are $\mathbf{w}$. With this property, we can rewrite the bilinear term as follows:

$$\begin{aligned} \mathbf{v}_i^{\mathrm{T}}\mathbf{W}\mathbf{v}_j &= \mathbf{v}_i^{\mathrm{T}}\mathbf{O}\,\mathrm{diag}(\mathbf{w})\mathbf{O}^{\mathrm{T}}\mathbf{v}_j \\ &= \mathbf{v}_i'^{\mathrm{T}}\,\mathrm{diag}(\mathbf{w})\mathbf{v}_j' \\ &= \langle \mathbf{v}_i', \mathbf{w}, \mathbf{v}_j' \rangle. \end{aligned} \tag{4.1}$$

where, $\mathbf{v}_i' = \mathbf{O}^{\mathrm{T}}\mathbf{v}_i$ and $\mathbf{v}_j' = \mathbf{O}^{\mathrm{T}}\mathbf{v}_j$, assuming $\mathbf{O}$ is learned implicitly. $\langle \mathbf{v}_i', \mathbf{w}, \mathbf{v}_j' \rangle$ is a "triple inner product" of $\mathbf{v}_i', \mathbf{w}$ and $\mathbf{v}_j'$ defined by $\langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle = \sum_{k=1}^{n} a_k b_k c_k$.

Consequently, imposing this symmetry constraint on the matrix can reduce the number of weight parameters from n^2 to n and both time and space complexity from $O(n^2)$ to $O(n)$.

Simultaneous Diagonalization

When a set of symmetric matrices forms a commuting family, they can be diagonalized by the same orthogonal matrix [11]. So we assume the weight matrices $\mathbf{U}^{[1]}, \mathbf{U}^{[2]}, \ldots, \mathbf{U}^{[L]}$ of the scoring functions (3.4) form a commuting family. Namely, we assume:

$$\mathbf{U}^{[p]}\mathbf{U}^{[q]} = \mathbf{U}^{[q]}\mathbf{U}^{[p]}, \ \forall p, q \in \{1, 2, \ldots, L\}.$$

With this assumption, all L weight matrices can be diagonalized simultaneously. Therefore, the vectors $\mathbf{v}_i^{label-head}$ and $\mathbf{v}_j^{label-dep}$ can be mapped by the same orthogonal matrix for all score functions.

Score Functions

Based on the above, under the assumption that all weight matrices are symmetric, we substitute the bilinear term in the biaffine transformations with a triple inner product. First, the score function for arcs is defined as follows:

$$\begin{aligned} s_{i,j}^{(arc)} &= \langle \mathbf{v}_i^{arc-head}, \mathbf{w}, \mathbf{v}_j^{arc-dep} \rangle \\ &\quad + (\mathbf{v}_i^{arc-head} \oplus \mathbf{v}_j^{arc-dep})^{\mathrm{T}} \mathbf{b}. \end{aligned} \tag{4.2}$$

where, $\mathbf{w} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^{2n}$. Unlike (3.3), the second term of this function contains the arc-dep vector because our experiments confirm that it improves performance.

Scoring functions for labels are defined as follows:

$$\begin{aligned} s_{i,j}^{(l)} &= \langle \mathbf{v}_i^{label-head}, \mathbf{u}^{[l]}, \mathbf{v}_j^{label-dep} \rangle \\ &\quad + (\mathbf{v}_i^{label-head} \oplus \mathbf{v}_j^{label-dep})^{\mathrm{T}} \mathbf{b}^{[l]}. \end{aligned} \tag{4.3}$$

where, $\mathbf{u}^{[l]} \in \mathbb{R}^m$, $\mathbf{b}^{[l]} \in \mathbb{R}^{2m}$. We eliminate the bias term $b^{[l]}$ in (3.4) because our experiments confirm that it does not affect performance.

4.2 Circulant Matrix Constraint

[15] used a circulant matrix for the bilinear transformation in a knowledge graph completion model [14] to reduce the number of parameters and improve the computational efficiency. Following this method, we assume the weight matrices of the bilinear term in the biaffine transformations are circulant and propose new scoring functions based on that.

Bilinear Transformation Using a Circulant Matrix

We define the circulant matrix $C(\mathbf{w}) \in \mathbb{R}^{n \times n}$ for a vector $\mathbf{w} \in \mathbb{R}^n$ as follows:

$$C(\mathbf{w}) = \begin{bmatrix} w_1 & w_n & \dots & w_3 & w_2 \\ w_2 & w_1 & w_n & & w_3 \\ \vdots & w_2 & w_1 & \ddots & \vdots \\ w_{n-1} & & \ddots & \ddots & w_n \\ w_n & w_{n-1} & \dots & w_2 & w_1 \end{bmatrix}. \tag{4.4}$$

where, $\mathbf{w}^{\mathrm{T}} = (w_1, \dots, w_n)$ and $c_{i,j} = w_{j-i(mod\ n)}$ ($c_{i,j}$ is the element at the ith row and the jth column of the matrix $C(\mathbf{w})$). Then, we replace the bilinear term with one where the weight matrix is a circulant matrix $C(\mathbf{w})$ with n parameters:

$$\mathbf{v}_i^{\mathrm{T}} C(\mathbf{w}) \mathbf{v}_j. \tag{4.5}$$

Score Functions

We propose score functions that employ (4.5) as the bilinear term in the biaffine transformation. The score function for an arc is then defined as follows:

$$\begin{aligned} s_{i,j}^{(arc)} &= \mathbf{v}_i^{arc-head} C(\mathbf{w}) \mathbf{v}_j^{arc-dep} \\ &\quad + (\mathbf{v}_i^{arc-head} \oplus \mathbf{v}_j^{arc-dep})^{\mathrm{T}} \mathbf{b}. \end{aligned} \tag{4.6}$$

where $\mathbf{w} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^{2n}$.

The score functions for labels are defined as follows:

$$\begin{aligned} s_{i,j}^{(l)} &= \mathbf{v}_i^{label-head} C(\mathbf{u}^{[l]}) \mathbf{v}_j^{label-dep} \\ &\quad + (\mathbf{v}_i^{label-head} \oplus \mathbf{v}_j^{label-dep})^{\mathrm{T}} \mathbf{b}^{[l]}. \end{aligned} \tag{4.7}$$

where, $\mathbf{u}^{[l]} \in \mathbb{R}^m$ and $\mathbf{b}^{[l]} \in \mathbb{R}^{2m}$.

Efficient Computation Using Fast Fourier Transform

In this section, we explain how to compute (4.5) efficiently using a fast Fourier transformation (FFT). We denote an n-point discrete Fourier transformation (DFT) matrix as $\mathfrak{F}_n \in \mathbb{C}^{n \times n}$. Then, any circulant matrix $C(\mathbf{w}) \in \mathbb{R}^{n \times n}$ can be diagonalized as follows [7]:

$$C(\mathbf{w}) = \mathfrak{F}_n^{-1} \operatorname{diag}(\mathfrak{F}_n \mathbf{w}) \mathfrak{F}_n.$$

With this property, we can rewrite (4.5) as a triple hermitian inner product [11]:

$$\begin{aligned} \mathbf{v}_i^{\mathrm{T}} C(\mathbf{w}) \mathbf{v}_j &= \mathbf{v}_i \mathfrak{F}_n^{-1} \operatorname{diag}(\mathfrak{F}_n \mathbf{w}) \mathfrak{F}_n \mathbf{v}_j \\ &= \frac{1}{n} \overline{\mathfrak{F}_n \mathbf{v}_i}^{\mathrm{T}} \operatorname{diag}(\mathfrak{F}_n \mathbf{w}) \mathfrak{F}_n \mathbf{v}_j \\ &= \langle \mathbf{v}_i', \mathbf{w}', \mathbf{v}_j' \rangle \\ &= \Re(\langle \mathbf{v}_i', \mathbf{w}', \mathbf{v}_j' \rangle). \end{aligned} \tag{4.8}$$

Here, $\mathbf{v}_i' = \overline{\mathfrak{F}_n \mathbf{v}_i}$, $\mathbf{v}_j' = \mathfrak{F}_n \mathbf{v}_j$, and $\mathbf{w}' = \frac{1}{n} \operatorname{diag}(\mathfrak{F}_n \mathbf{w})$, and all are n dimensional complex vectors. $\Re(\cdot)$ is the operator which takes the real parts of its argument. With this transformation, we can compute the bilinear transformation using a circulant matrix in $O(n \log n)$ using a FFT.

The DFT of an n-dimensional vector $\mathbf{x}$, $\mathfrak{F}_n \mathbf{x}$, is conjugate symmetric if and only if $\mathbf{x}$ is a real vector [9]. In our experiment, we initialize $\mathbf{w}'$ with the DFT of a real vector and update it in complex space. We update "frequency" domain (complex space) vectors using only the operations which have correspondence to "time domain" vectors. Thus, as described in [9], the conjugate symmetry of the vectors are maintained during learning because their initial values satisfy the condition.

Expressiveness of the Bilinear Transformation Using Circulant Matrices

In this section, we explain the expressiveness of circulant matrices in relation to an arbitrary matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. [19] show that for any $\mathbf{W} \in \mathbb{R}^{n \times n}$, there exists a

normal matrix $\mathbf{W}' \in \mathbb{C}^{n \times n}$ such that $\mathbf{W} = \Re(\mathbf{W}')$. Further, as with a symmetric matrix, a normal matrix can be diagonalized as follows:

$$\mathbf{W} = \Re(\mathbf{W}') = \Re(\mathbf{O}\,\mathrm{diag}(\mathbf{w}')\mathbf{O}^*).$$

Here, $\mathbf{O} \in \mathbb{C}^{n \times n}$ is a unitary matrix, $\mathbf{O}^*$ is the conjugate transpose of $\mathbf{O}$, and $\mathbf{w}' \in \mathbb{C}^n$ is the complex vector which consists of the eigenvalues of $\mathbf{W}'$. By replacing the weight matrix $\mathbf{W}$ in the first term of the bilinear transformation with (4.2), that term can be expressed as (4.8). The unitary matrix $\mathbf{O}$ is a bijective function, so the input vectors $\mathbf{v}_i$, $\mathbf{v}_j$ are learned as their one-to-one correspondents $\mathbf{v}'_i = \mathbf{O}^{\mathrm{T}}\mathbf{v}_i$, $\mathbf{v}'_j = \mathbf{O}^{\mathrm{T}}\mathbf{v}_j$, assuming that the unitary matrix $\mathbf{O}$ is learned implicitly. Note that to simultaneously diagonalize the normal matrices whose real parts are the weight matrices $\mathbf{U}^{[1]}, \mathbf{U}^{[2]}, \ldots, \mathbf{U}^{[L]}$ in the scoring functions for labels, we have to assume that they form a commuting family as with the discussion in 4.1.

5 Experiments

5.1 Dataset and Implementation

We compared the models described in the previous chapter on several languages in the CoNLL 2017 shared task for Universal Dependency Parsing dataset. We chose four languages which have relatively abundant training examples: UD_Chinese, UD_Czech, UD_English and UD_German. In addition, we selected four languages which have fewer training examples: UD_French-ParTUT, UD_Galician-TreeGal, UD_Latin and UD_Slovenian-SST.

As a baseline model, we used Timothy Dozat's dependency parser*, which achieved the highest accuracy on the shared task. The structures of the proposed models are based on the baseline model; we changed only the classifier part.

We only modified two hyper-parameters: We used no pretrained embeddings and initialized word embeddings with a uniform distribution. These settings remain the same throughout all experiments unless otherwise stated.

Although word segmentation and POS tagging are included in the shared task, we use the gold word segmentation and gold POS tags for our models. We excluded these two tasks because they are irrelevant to the objective of this research, which is to show the effects of the proposed methods on biaffine classifiers.

5.2 Results

We show the results of the baseline and our two proposed methods in Table 5.1. Here, Unlabeled Attachment Score (UAS) assesses if the model can find correct head-dependent pairs. In addition to UAS, Labeled Attachment Score (LAS) evaluates the model's ability to predict the syntactic relations for those pairs.

*https://github.com/tdozat/Parser-v2

Treebank	Baseline		Symmetric Matrix		Circulant Matrix	
	UAS	**LAS**	**UAS**	**LAS**	**UAS**	**LAS**
UD_Czech	93.72	91.89	93.45(-0.27)	91.50(-0.39)	**93.87(+0.15)**	**92.01(+0.12)**
UD_German	87.57	84.27	87.21(-0.36)	83.91(-0.36)	**87.61(+0.04)**	**84.39(+0.12)**
UD_English	**91.05**	**89.42**	90.95(-0.1)	89.22(-0.2)	91.04(-0.01)	89.31(-0.11)
UD_Chinese	87.67	85.58	87.55(-0.12)	85.41(-0.17)	**87.96(+0.29)**	**85.65(+0.07)**
UD_Slovenian-SST	75.63	69.53	74.94(-0.69)	68.58(-0.95)	**75.90(+0.27)**	**70.24(+0.71)**
UD_Latin	70.90	64.53	70.27(-0.63)	63.18(-1.35)	**72.38(+1.48)**	**66.00(+1.47)**
UD_French-ParTUT	91.82	89.78	**92.09(+0.27)**	89.94(+0.16)	91.94(+0.12)	**90.09(+0.31)**
UD_Galician-TreeGal	80.10	74.77	80.05(-0.05)	74.78 (+0.01)	**80.24(+0.14)**	**75.68(+0.91)**

Table 5.1: Main results on CoNLL 2017 dataset.

The method based on circulant matrices outperformed the others on almost all languages except for English, where the baseline model achieved the best accuracy, and French-ParTUT, where the method based on symmetric matrices outperformed it on UAS. Interestingly, the method based on symmetric matrices underperformed the baseline on most languages. This might be because of the restricted expressiveness of a symmetric weight matrix in comparison to a circulant one, especially the former's inability to express asymmetric relations.

6 Analysis

In this section, we examine the robustness to overfitting of the proposed methods.

6.1 Robustness on Smaller Training Data

First, to further examine how the number of parameters affect the models, we conducted additional experiments on the UD_English treebank, reducing the number of training examples by a quarter each time.

The results of this experiment are shown in Table 6.1. Our methods performed better on smaller datasets, where the number of training examples was less than or equal to half the original number of examples. This indicates that our methods achieved high generalizability through parameter reduction.

Second, we ran the baseline model with classifier dimensions reduced from 400 to 200 for the arc classifier and 100 to 50 for the label classifier and compared it with the proposed methods on languages with small treebanks. As shown in Table 6.2, simply reducing the number of dimensions hindered the accuracies in some languages, while the method based on circulant matrices consistently outperformed the baseline model with the original number of dimensions in all of these languages with small treebanks, as shown in Table 6.2. This result indicates the effectiveness of the proposed method on small datasets.

6.2 Parameter Reduction

Table 6.3 shows the proportion of the total parameters taken up by each of the components. While LSTMs occupy the largest portion of the model, the second largest is occupied by the classifiers, which account for about 17% of the

Reduced Samples	Baseline		Symmetric Matrix		Circulant Matrix	
	UAS	**LAS**	**UAS**	**LAS**	**UAS**	**LAS**
0 / 4	**91.05**	**89.42**	90.95(-0.1)	89.22(-0.2)	91.04(-0.01)	89.31(-0.11)
1 / 4	**90.32**	**88.57**	90.05(-0.27)	88.30(-0.27)	90.29(-0.03)	88.49(-0.08)
2 / 4	88.98	87.08	**89.15(+0.17)**	**87.17(+0.09)**	88.72(-0.26)	86.63(-0.45)
3 / 4	87.24	85.08	87.27(+0.03)	85.06(-0.02)	**87.59(+0.35)**	**85.38(+0.3)**

Table 6.1: Results in UD_English with fewer training samples.

Treebank	**UAS**	**LAS**
UD_Slovenian-SST	74.98(-0.65)	69.07(-0.46)
UD_Latin	71.96(+1.06)	65.68(+1.15)
UD_French-ParTUT	92.17(+0.35)	90.44(+0.66)
UD_Galician-TreeGal	79.68(-0.42)	74.82(+0.05)

Table 6.2: Baseline model with reduced dimensions. The numbers in parentheses are the differences from the baseline model with full dimensions.

parameters. Table 6.4 indicates that the proposed methods were able to reduce the number of parameters by more than 16%.

Although we showed that our proposed methods reduced the number of parameters by more than 16% compared to the baseline model, this result does not include any word embeddings. We did not include word embeddings in Table 6.3 because the vocabulary size varies across treebanks and is not dependent on the structure of the model. However, this portion is not small enough to neglect for real applications. Since we used 100 dimensional word embeddings, the total number of parameters required for them is 100 $\times$ vocabulary size. For example, in the case of UD_English, since the vocabulary size in the training data is 16653, the number of parameters required for word embeddings is 100 $\times$ 16653, which accounts for 34.53% of the total parameters of the baseline model. Factoring this in, our proposed methods reduce the number of parameters by 11%.

	Number of Parameters	Percentage
Character LSTM	241200	7.64%
Bidirectional LSTM	1927200	61.03%
Arc MLP	320800	10.16%
Label MLP	80200	2.54%
Arc Classifier	160400	5.08%
Label Classifier	377437	11.95%
Others	50400	1.60%
TOTAL	3157637	100%

Table 6.3: Percentage of baseline model parameters accounted for by each component.

	Baseline	Symmetric Matrix	Circulant Matrix
Arc Classifier	160400	1200	1600
Label Classifier	377437	11100	14800
Sum with shared parts	3157637	2632100	2636200
Difference from the baseline	0.0%	**-16.64%**	**-16.51%**

Table 6.4: Comparison of parameter sizes.

6.3 Parsing Speed

To test the parsing speed, we used an NVidia GTX1080 GPU and parsed the test dataset of UD_English. As mentioned in Section 4, both proposed methods are superior to the baseline model in terms of time complexity. Specifically, the methods based on symmetry took 13.86 seconds, circularity 15.06 seconds, and the baseline 17.76 seconds. These results are in accordance with the theoretical time complexity.

7 Conclusion

In this paper, we reduced the number of parameters in the weight matrices of biaffine classifiers based on assumptions of symmetry or circularity and examined the effects of these reductions on the CoNLL 2017 shared task for the Universal Dependency Parsing dataset.

As a result, the method based on circulant matrices outperformed the baseline model in most languages, and the method based on symmetric matrices achieved the highest UAS on UD_French-ParTUT. Furthermore, both of the proposed models achieved more than a 16% parameter reduction. One of the possible reasons for the improvement of the parsing performance is the reduction of the expressiveness of the weight matrices. That is, the assumption that the matrices formed commutative families enabled the models to learn more efficiently by restricting the parameter search space.

As future work, the L1 regularization method for CompLex [18] proposed by [13] may be integrated into our methods to further reduce the number of parameters in the bilinear function. The script for the bilinear functions proposed here is provided on the GitHub page below*.

*https://github.com/TomokiMatsuno/PACLIC32/blob/master/my_linalg.py

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL `http://arxiv.org/abs/1409.0473`.

[2] Chu and Liu. On the shortest arborescence of a directed graph. *Science Sinica*, Vol.14, 1965.

[3] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal stanford dependencies: A cross-linguistic typology. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *LREC*, pages 4585–4592. European Language Resources Association (ELRA), 2014. URL `http://dblp.uni-trier.de/db/conf/lrec/lrec2014.html#MarneffeDSHGNM14`.

[4] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734, 2016. URL `http://arxiv.org/abs/1611.01734`.

[5] Timothy Dozat, Peng Qi, and Christopher D Manning. Stanford's graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, 2017.

[6] Jack Edmonds. Optimum branchings. *JOURNAL OF RESEARCH of the National Bureau of Standards*, 71(4), 1967.

[7] Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006.

[8] Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/D17-1206`.

[9] Katsuhiko Hayashi and Masashi Shimbo. On the equivalence of holographic and complex embeddings for link prediction. In Regina Barzilay and Min-Yen Kan, editors,

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers, pages 554–559. Association for Computational Linguistics, 2017. ISBN 978-1-945626-76-0. doi: 10.18653/v1/P17-2088. URL https://doi.org/10.18653/v1/P17-2088.

[10] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016. URL http://aclweb.org/anthology/Q16-1023.

[11] Hanxiao Liu, Yuexin Wu, and Yiming Yang. Analogical inference for multi-relational embeddings. *CoRR*, abs/1705.02426, 2017. URL http://arxiv.org/abs/1705.02426.

[12] X. Ma, Z. Hu, J. Liu, N. Peng, G. Neubig, and E. Hovy. Stack-Pointer Networks for Dependency Parsing. *ArXiv e-prints*, May 2018.

[13] Hitoshi Manabe, Katsuhiko Hayashi, and Masashi Shimbo. Data-dependent learning of symmetric/antisymmetric relations for knowledge base completion. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16211.

[14] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816, 2011.

[15] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. Holographic embeddings of knowledge graphs. *CoRR*, abs/1510.04935, 2015. URL http://arxiv.org/abs/1510.04935.

[16] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

[17] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[18] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2071–2080. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045609.

[19] Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *Journal of Machine Learning Research*, 18:130:1–130:38, 2017. URL `http://jmlr.org/papers/v18/papers/v18/16-563.html`.

[20] Daniel Zeman. Reusable tagset conversion using tagset drivers. In *LREC*. European Language Resources Association, 2008. URL `http://dblp.uni-trier.de/db/conf/lrec/lrec2008.html#Zeman08`.

[21] Daniel Zeman and Jan Hajic, editors. *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, October 31 - November 1, 2018*, 2018. Association for Computational Linguistics. ISBN 978-1-948087-82-7.

[22] Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19. Association for Computational Linguistics, 2017. doi: 10.18653/v1/K17-3001. URL `http://www.aclweb.org/anthology/K17-3001`.

Publication List

International Conference (Refereed)

[1] Tomoki Matsuno, Katsuhiko Hayashi, Takahiro Ishihara, Hitoshi Manabe, Yuji Matsumoto. Reduction of Parameter Redundancy in Biaffine Classifiers with Symmetric and Circulant Weight Matrices. The 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32), December 2018.

Other Publications (Non-Refereed)

[1]松野智紀, 能地宏, 松本裕治. ニューラルグラフ型係り受け解析器のための文節ベクトル表現. 言語処理学会第24回年次大会発表論文集, pp.600-603, March 2018
[2] 松野智紀, 林克彦 (阪大), 石原敬大, 真鍋陽俊 (ワークスアプリケーションズ), 松本裕治. 重み行列の対称性および巡回性の仮定に基づく双アフィン分類器の冗長性削減. 情報処理学会第236回自然言語処理研究会研究報告自然言語処理（NL）,2018-NL-236(8).
[3]松野智紀, 松本裕治. 単語-文節間の変換による日本語係り受け解析. 言語処理学会第25回年次大会, March 2019 (to appear)