チョムスキー階層とニューラル言語モデル

染谷大河　吉田遼　中石海　大関洋平
東京大学
{taiga98-0809, yoshiryo0617, nakaishi-kai787, oseki}@g.ecc.u-tokyo.ac.jp

概要

近年、自然言語の文を用いて、言語モデルがどのような言語現象を把握できるかが盛んに検証されている。また、形式言語を用いて、言語モデルがチョムスキー階層のどのクラスに属する言語までを認識できるのかも検証されてきている。しかしながら、自然言語を用いた研究では、同種の統語構造を持った言語現象を抽象化して統一的に扱うという観点が欠けていたために、語彙の影響と統語構造の影響を切り分けることが十分にできておらず、また形式言語を用いた研究では、終端記号の種類数が最低限の場合しか扱われておらず、自然言語のように多様な終端記号が存在する場合でも同様に認識できるのかという点は検討されてこなかった。そこで本研究では、自然言語を抽象化し語彙の影響を排除したモデルとして多様な終端記号を持つ形式言語を考え、ニューラル言語モデルがこのような言語をどの程度認識できるか実験する。

1　導入

ニューラル言語モデルはどの程度統語的に複雑な言語現象を捉えられるのだろうか。近年、ニューラル言語モデルの成功に伴い、その統語的評価が盛んに行われている [1, 2, 3, 4]。これら統語的評価に関する先行研究では、実際に自然言語の文を用いてニューラル言語モデルがどのような言語現象を把握できるのかが検証されており、例えば、これら研究の先駆けとなった Linzen et al. (2016) [1] では、再帰的ニューラルネットワーク (Recurrent Neural Network, RNN; Elman, 1990 [5]) をベースとしたモデルである Long Short-Term Memory (LSTM; Hochreiter and Schmidhuber, 1997 [6]) 言語モデルが英語における主語と動詞の一致を捉えられることが示されている。また、より近年では、様々なタスクで世界最高性能を達成している Transformer [7] をベースとしたモデルである GPT-2 [8] 言語モデルが、LSTM 言語モデルよりも高い精度で主語と動詞の一致・島の制約などの様々な言語現象を捉えられること [9] も示されている。しかしながら、これらの研究では、主にニューラル言語モデルが個別の言語現象に対して汎化できるかが検証されるにとどまり、同種の統語構造を持つ言語現象を統一的に扱うという観点が欠けていた。そのため、統語構造の複雑さが本質的にどのように結果に影響しているかが不明瞭だった。また、検証には実際の自然言語を用いているため、語彙の影響を排除できず、言語現象を統語的な複雑性に注目して比較することができないという方法論的な限界があった。

一方、先行研究では、ニューラル言語モデルがどの程度統語的に複雑な「言語」を認識できるのかも盛んに議論されてきている [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]。これらの先行研究では、ニューラル言語モデルがチョムスキー階層のどのクラスに属する形式言語までを認識できるのかが、数理的・実験的に検証されてきており、例えば、RNN や LSTM はいくつかの文脈自由言語を認識できること [24, 25, 15] や、Transformer はチョムスキー階層にうまく位置付けられないこと [17, 19] などが示唆されている。しかしながら、これらの研究は、最低限の数の終端記号を持つ形式言語での検証にとどまっており、同クラスに属する言語であっても、自然言語のように多様な終端記号を持つ場合に、ニューラル言語モデルが認識できるとは限らないという限界があった。

そこで本研究では、自然言語を多様な終端記号を持つ形式言語に抽象化し、語彙の影響を排除して統語的な複雑さに注目することで、ニューラル言語モデルがどの程度統語的に複雑な言語現象を捉えられるのかを検証する。具体的には、英語における複合語などの文法現象を A^nB^n 型の形式言語、ドイツ語における依存関係などの文法現象を Nested Dependency 型の形式言語、Swiss German における依存関係などの文法現象を Cross Serial Dependency 型

の形式言語に抽象化し、LSTM・GPT-2 言語モデルのそれぞれがこれらのうちどの程度複雑なものまで認識できるのかを検証する。

検証の結果、LSTM 言語モデルは A^nB^n 型の言語現象については認識することができ、高い精度で学習の際よりも長い文字列に汎化することが可能だが、Nested Dependency 型や Cross Serial Dependency 型の形式言語はほとんど全く汎化できないことがわかった。一方 GPT-2 言語モデルは、いずれの言語においても、LSTM が A^nB^n 型に対して達成したほど高い精度での汎化は不可能であった。また、いずれの言語モデル、形式言語においても、終端記号の種類を増やしていくと精度が下がる傾向が見られた。

2 実験

2.1 扱う形式言語の種類

本研究で扱う言語はいずれも以下のように定義される：有限個の非終端記号の集合 $V_{\text{non.}}$ を与える。各非終端記号 $A \in V_{\text{non.}}$ に対して、T 個の終端記号 $a_{A,0}, \cdots, a_{A,T}$ を定義する。非終端記号のみからなる有限長の文字列の集合 $L_{\text{non.}} \subseteq V^*_{\text{non.}}$ を決める。$L_{\text{non.}}$ に含まれる任意の文字列をとり、それを構成する各非終端記号 A を $a_{A,0}, \cdots, a_{A,T}$ のいずれかに書き換える。ここで、文字列に同じ非終端記号が複数個含まれる場合、それらを異なる終端記号に書き換えても良い。こうして得られる終端記号のみの文字列全体の集合を言語 L とする。言語 L は、非終端記号の集合 $V_{\text{non.}}$、各非終端記号に対応する終端記号の個数 T、そして非終端記号の文字列の集合 $L_{\text{non.}}$ を与えると決まる。

以下、本研究で扱う言語を順に導入し、それぞれの Chomsky 階層における位置付け、および自然言語との対応について述べる。

2.1.1 A^nB^n 型

$$V_{\text{non.}} = \{A, B\}, \tag{1}$$
$$L_{\text{non.}} = \{A^nB^n : n \geq 0\} \tag{2}$$

で定義される文脈自由言語。この言語に対応する自然言語 $T \geq 2$ の例としては、オランダ語などに見られる (Ⅰ) のような構造が挙げられる [26]。

ブラケット内には NP（名詞句）と V（動詞）が３つずつ含まれ、NP の個数と V の個数が一致している必要がある。一方、各 NP と V のあいだに文法的な対応関係は特に要求されない。これは、書き換え規則を $A \to$ Marie|Pieter|Arabisch|$\cdots$ および $B \to$ laat|zien|schrijven|$\cdots$ とした場合に対応する。

(Ⅰ) dass Jan [Marie Pieter Arabisch laat zien schrijven]
that Jan Marie Pieter Arabic let see write
'that Jan let Marie see Pieter write Arabic'

2.1.2 Nested Dependency 型

$$V_{\text{non.}} = \{A_0, \cdots, A_{N-1}, B_0, \cdots, B_{N-1}\}, \tag{3}$$
$$L_{\text{non.}} = \{A_{i_0} \cdots A_{i_{n-1}} B_{i_{n-1}} \cdots B_{i_0} : n \geq 0;\ 0 \leq i_0, \cdots, i_{n-1} \leq N-1\} \tag{4}$$

で定義される文脈自由言語。ここで、同じ添字 i をもつ A_i と B_i は文法的な対応関係を持つことを表すと考えると、自然言語では英語の以下の (Ⅱ) のような構造と対応する [27]。この構造では複数名詞句 the cats と複数動詞 bark が対応し、単数名詞句 the dog と単数動詞 chases がそれぞれ対応している。このように文法数（単数/複数）を持つ動詞の数とそれに対応する文法数を持つ名詞句の数が一致する。よって、非終端記号を $A_0 = \text{V}_{\text{単数}}, A_1 = \text{V}_{\text{複数}}, \cdots$、$B_0 = \text{NP}_{\text{単数}}, B_1 = \text{NP}_{\text{複数}}, \cdots$ とし、終端記号への書き換え規則を $\text{V}_{\text{単数}} \to$ chases|barks|$\cdots$、$\text{V}_{\text{複数}} \to$ chase|bark|$\cdots$、$\text{NP}_{\text{単数}} \to$ the cat|the dog|$\cdots$、$\text{NP}_{\text{複数}} \to$ the cats|the dogs|$\cdots$ とした場合に対応する。

(Ⅱ) the cats that the dog chases bark

2.1.3 Cross Serial Dependency 型

$$V_{\text{non.}} = \{A_0, \cdots, A_{N-1}, B_0, \cdots, B_{N-1}\}, \tag{5}$$
$$L_{\text{non.}} = \{A_{i_0} \cdots A_{i_{n-1}} B_{i_0} \cdots B_{i_{n-1}} : n \geq 0;\ 0 \leq i_0, \cdots, i_{n-1} \leq N-1\} \tag{6}$$

で定義される文脈依存言語 [28]。ここで、同じ添字 i をもつ A_i と B_i は文法的な対応関係を持つことを表すと考えると、自然言語では、Swiss German における以下の (Ⅲ) のような構造に対応する [29]。この構造では、与格動詞 hälfe は与格名詞句 em Hans を、対格動詞 aastriiche が対格名詞句 es huus を項としている。このように文法格を持つ

動詞の数とそれに対応する格を持つ名詞句の数が一致する。さらに、対応する動詞と名詞句は格が一致していなければならない。よって、非終端記号を $A_0 = \mathrm{V}_{与格}, A_1 = \mathrm{V}_{対格}, \cdots$、$B_0 = \mathrm{NP}_{与格}, B_1 = \mathrm{NP}_{対格}, \cdots$ とし、終端記号への書き換え規則を $\mathrm{V}_{与格} \rightarrow$ hälfe$|\cdots$、$\mathrm{V}_{対格} \rightarrow$ aastriiche$|\cdots$、$\mathrm{NP}_{与格} \rightarrow$ em Hans$|$em huus$|\cdots$、$\mathrm{NP}_{対格} \rightarrow$ de Hans$|$es huus$|\cdots$ とした場合に対応する。

(III) ... mer em Hans es huus hälfe aastriiche
... we Hans the house help paint.
'... we help Hans paint the hause'

2.2 データ生成

本研究では、Nested Dependency 型と Cross Serial Dependency 型については $V_{\text{non.}} = \{A_0, \cdots, A_4, B_0, \cdots, B_4\}$ の場合のみで検証する。そして、各形式言語について $T = 2, 5, 10, 100, 1000, 5000$ の 6 つの場合を考え、合計で 18 種類の形式言語を言語モデルが認識できるかを検証する。

以上の 18 種類の各文法によって生成される文のうち、長さが $l \sim \mathcal{U}(4, 30)$ である文を 50,000 文サンプリングして作成し、そのうち 40,000 文を学習用データ、5,000 文ずつを検証・テスト用データとした。また、各文法から生成される文のうち、長さが $l \sim \mathcal{U}(31, 100)$ である文を 5,000 文をサンプリングし、「汎化テスト用データ」とした。これは、言語モデルが学習時よりも長い文字列に対してその文字列が当該言語に含まれるかどうかの判定を正しくできるかどうかを検証するためのものである。最後に、以上により生成した 55,000 文の正例それぞれに対して以下の方法で対応する負例を作成して追加し、各形式言語それぞれに対して合計 110,000 文ずつからなるデータセットを生成した：A^nB^n 型の形式言語については、それぞれの正例に含まれる B の数 n を $m \sim \mathcal{U}(min\_sequence\_length, max\_sequence\_length)$ に変化させることによって負例を生成した。ただし、$n = m$ となった場合は $n \neq m$ となるまでサンプリングを繰り返した。また、ここで $min\_sequence\_length$ と $max\_sequence\_length$ は各データに含まれる文の長さの最小値と最大値である。また、Nested Dependency 型と Cross Serial Dependency 型の形式言語については、それぞれの正例に含まれる N 個の B_n $(0 \leq n \leq N-1)$ のうち、$l \sim \mathcal{U}(1, N-1)$ 個を B_m $(0 \leq m \leq N-1, m \neq n)$ に入れ替えることで負例を生成した。本研究では、データセット内の各文についてそれが正例であるか負例であるかの二値分類を行うタスクを考え、その正解率で言語モデルの性能を評価する。

2.3 言語モデル

本研究では、前節で作成されたデータを用いて二種類のニューラル言語モデルの性能を評価する。

LSTM 本研究では、PyTorch[1] で実装された、単語埋め込み次元が 256、隠れ層の次元が 256 の 1 層 LSTM 言語モデル [6] を使用する。

GPT-2 本研究では、Huggingface[2] により実装された 3 層、4 ヘッド、単語埋め込み次元が 128 の GPT-2 言語モデル [8] を使用する。

言語モデルの学習 LSTM 言語モデルは、最適化アルゴリズムとして SGD を用いた。GPT-2 言語モデルは、最適化アルゴリズムとして AdamW [30] を用い、その他のハイパーパラメータはデフォルト値を用いた。両言語モデルともにバッチサイズは 512 で 15 エポック訓練し、検証用データでの損失が最も小さくなった時点のモデルを評価した。

3 結果と考察

表 1 は各言語モデルの各タスクに対する汎化テスト用データでの正解率である。先行研究 [23] に倣い、各タスクに対する正解率は、異なる 10 のランダムシード、異なる 3 つの学習率を用いて学習された言語モデルの正解率の最大値とし、正解率が 90% 以上となった場合に言語モデルがそのタスクを解くことができたと判断することとする。

LSTM 言語モデルは、終端記号の種類が最低限の場合、即ち $T = 1$ の場合には、A^nB^n 型のタスクを解くことができるとことが先行研究で実験的に確かめられていたが [13]、本研究でも、LSTM 言語モデルは A^nB^n 型のタスクを $T \leq 100$ の場合で解くことが可能だと示された。これは、先行研究で確かめられていた予想が、一定程度まで終端記号を増やしていっても成り立つことを示唆している。一方で、Nested Dependency 型や Cross Serial Dependency 型のタスクの正解率は A^nB^n 型の場合のように一貫して高くなく、LSTM がこれらの統語構造を必ずしも認識できないことを示唆する。特に、Nested dependency 型は

1) https://pytorch.org/
2) https://huggingface.co/

タスク	終端記号 (T)	LSTM	GPT-2
A^nB^n	2	**100.0**	55.70
	5	**100.0**	57.80
	10	**100.0**	56.39
	100	**100.0**	69.64
	1000	52.17	62.18
	5000	53.82	50.85
Nested Dependency	2	**96.15**	87.13
	5	86.83	**92.29**
	10	80.35	**94.41**
	100	50.89	50.95
	1000	50.91	50.73
	5000	51.17	50.83
Cross Serial Dependency	2	89.51	81.95
	5	84.55	83.13
	10	72.61	**92.94**
	100	51.70	52.27
	1000	50.67	50.86
	5000	51.12	51.01

表 1　言語モデルの各タスクに対する汎化テスト用データでの正解率。正解率は、異なる 10 のランダムシード、異なる 3 つの学習率を用いて学習された言語モデルの正解率の最大値をとったものである。正解率が 90% を超えたものを太字で示している。

A^nB^n 型と同じ文脈自由言語であり、LSTM が前者は認識できず後者は認識できることは、Chomsky 階層において同じ階層に分類される形式言語であっても、ある程度終端記号を増やしても認識できるものと、少ない終端記号数であっても認識できないものがあることを意味する。同様に、Cross Serial Dependency 型は文脈依存言語であるが、LSTM がこれを終端記号数にかかわらず認識できないことは、LSTM が $A^nB^nC^n$ 型の文脈依存言語を捉えることができるとする先行研究 [31] の結果と合わせると、Chomsky 階層において同じ階層に分類される形式言語であっても、終端記号数に応じて認識できるものと認識できないものがあることを意味する。

GPT-2 言語モデルは、Nested Dependency 型の一部 ($T = 5, 10$) や Cross Serial Dependency 型の一部 ($T = 10$) で 90%を超える正解率を達成しているものの、全ての場合で一貫してタスクを解くことはできなかった。これは、高い表現力を持つため言語現象の大局的なパターンを捉えることはできるが、生起回数を数えることや複雑な依存関係のルールを捉えることが必ずしもできていないことを示唆していると考えられる。一方で、言語モデルが自然言語にある個別の言語現象を理解できているかを自然言語を用いて検証した先行研究では、GPT-2 など Transformer ベースの言語モデルが LSTM などの再帰的ニューラルネットワーク言語モデルよりも幅広い言語現象を理解できることが示唆されている [9]。このことは一見すると本研究の結果と整合性が無いように思われるが、以下のように説明できる。これらの先行研究では、言語モデルが個別の言語現象をどの程度理解しているかを各論的に評価できる。しかし、言語モデルが自然言語の背後にある統語構造を認識せずに、語彙的なヒューリスティックを用いてタスクを解いている可能性を排除できない。一方、本研究では、統語構造に注目して語彙的要素を抽象化した形式言語を対象としており、これが本研究の結果との相違を生じさせている可能性がある。

また、いずれの言語モデル、タスクについても、終端記号の種類を増やすと正解率は下がる傾向が見られた。[13] や [23] などの先行研究では、終端記号の種類数が最低限であるような形式言語のみに対して言語モデルの性能が評価されていたが、本研究のこの結果は、これらの研究で認識可能とされていた形式言語が、終端記号の種類を増やすと認識不可能になる可能性を示唆する。自然言語では、終端記号に対応する単語などの種類は非常に多様である。従って、言語モデルが最低限の終端記号しか持たない形式言語を認識できたとしても、それは多様な終端記号を持つ自然言語を認識できることを意味しない。この意味で、本研究は、終端記号の種類数の影響を検証する必要性を提起する。

4　結論

本研究では、多様な終端記号への書き換え規則も含め自然言語を形式言語化し、ニューラル言語モデルがどの程度統語的に複雑な言語現象を捉えられるのかを検証した。検証の結果、LSTM 言語モデルは A^nB^n 型の言語現象を終端記号数が少ない場合に認識することができるが、Nested Dependency 型や Cross Serial Dependency 型の形式言語は終端記号数に関係なくほとんど全く汎化することができなかった。一方 GPT-2 言語モデルは、いずれの言語においても、LSTM が A^nB^n 型に対して達成したほど高い精度での汎化は不可能であった。また、いずれの場合でも終端記号の種類を増やしていくと精度が下がる傾向が見られた。対象となる言語クラスの範囲を拡大することは今後の課題としたい。

謝辞

本研究は JST さきがけ JPMJPR21C2 の支援を受けたものである。

参考文献

[1] Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn Syntax-Sensitive dependencies. **Transactions of the Association for Computational Linguistics**, Vol. 4, pp. 521–535, December 2016.

[2] Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 1192–1202, Brussels, Belgium, 2018. Association for Computational Linguistics.

[3] Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. What do RNN language models learn about Filler–Gap dependencies? In **Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP**, pp. 211–221, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[4] Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. Neural language models as psycholinguistic subjects: Representations of syntactic state. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 32–42, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] Jeffrey L Elman. Finding structure in time. **Cogn. Sci.**, Vol. 14, No. 2, pp. 179–211, March 1990.

[6] S Hochreiter and J Schmidhuber. Long short-term memory. **Neural Comput.**, Vol. 9, No. 8, pp. 1735–1780, November 1997.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L Ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30, pp. 5998–6008. Curran Associates, Inc., 2017.

[8] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[9] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. BLiMP: The benchmark of linguistic minimal pairs for english. **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 377–392, December 2020.

[10] Janet Wiles and Jeffrey Elman. Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. 06 1995.

[11] Paul Rodriguez and Janet Wiles. Recurrent neural networks can learn to implement symbol-sensitive counting. Vol. 10, , 1997.

[12] Luzi Sennhauser and Robert Berwick. Evaluating the ability of LSTMs to learn context-free grammars. pp. 115–124, November 2018.

[13] Gail Weiss, Yoav Goldberg, and Eran Yahav. On the practical computational power of finite precision RNNs for language recognition. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 740–745, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[14] Samuel A. Korsky and Robert C. Berwick. On the computational power of rnns. **CoRR**, Vol. abs/1906.06349, , 2019.

[15] Mirac Suzgun, Yonatan Belinkov, Stuart Shieber, and Sebastian Gehrmann. LSTM networks can perform dynamic counting. In **Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges**, pp. 44–54, Florence, August 2019. Association for Computational Linguistics.

[16] William Merrill. Sequential neural networks as automata. **CoRR**, Vol. abs/1906.01615, , 2019.

[17] Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the Ability and Limitations of Transformers to Recognize Formal Languages. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 7096–7116, Online, November 2020. Association for Computational Linguistics.

[18] Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How can self-attention networks recognize dyck-n languages? **CoRR**, Vol. abs/2010.04303, , 2020.

[19] Michael Hahn. Theoretical limitations of self-attention in neural sequence models. **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 156–171, 2020.

[20] William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. A formal hierarchy of RNN architectures. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 443–459, Online, July 2020. Association for Computational Linguistics.

[21] Joshua Ackerman and George V. Cybenko. A survey of neural networks and formal languages. **ArXiv**, Vol. abs/2006.01338, , 2020.

[22] Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers, 2021.

[23] Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Marcus Hutter, Shane Legg, and Pedro A. Ortega. Neural networks and the chomsky hierarchy. 2022.

[24] Paul Rodriguez and Janet Wiles. Recurrent neural networks can learn to implement symbol-sensitive counting. In M. Jordan, M. Kearns, and S. Solla, editors, **Advances in Neural Information Processing Systems**, Vol. 10. MIT Press, 1997.

[25] Natalia Skachkova, Thomas Trost, and Dietrich Klakow. Closing brackets with recurrent neural networks. In **Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP**, pp. 232–239, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[26] Geoffrey K. Pullum and Gerald Gazdar. Natural languages and context-free languages. **Linguistics and Philosophy**, Vol. 4, No. 4, pp. 471–504, 1982.

[27] Hana Filip. **LIN 69321 LIN6932 Topics in Computational Linguistics Lecture 7**. n.d.

[28] Alfred V Aho and Jeffrey D Ullman. **The Theory of Parsing, Translation and Compiling. Parsing, vol. I**. Prentice-Hall, Englewood Cliffs, 1972.

[29] Stuart M. Shieber. Evidence against the context-freeness of natural language. **Linguistics and Philosophy**, Vol. 8, No. 3, pp. 333–343, 1985.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019**. OpenReview.net, 2019.

[31] F.A. Gers and E. Schmidhuber. LSTM recurrent networks learn simple context-free and context-sensitive languages. **IEEE Transactions on Neural Networks**, Vol. 12, No. 6, pp. 1333–1340, January 2001. Conference Name: IEEE Transactions on Neural Networks.