UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Indução gramatical automática para o português

Diego Pedro Gonçalves da Silva

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Diego Pedro Gonçalves da Silva

Indução gramatical automática para o português

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Thiago Alexandre Salgueiro Pardo

USP – São Carlos
Julho de 2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

P372i

Pedro Gonçalves da Silva, Diego
Indução gramatical automática para o português / Diego Pedro Gonçalves da Silva; orientador Thiago Alexandre Salgueiro Pardo. -- São Carlos, 2024.
157 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024.

1. CE610.4.1. 2. CE610.27.3. 3. CH791.20. 4. CE610.4. 5. CH791.6. I. Alexandre Salgueiro Pardo, Thiago, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Diego Pedro Gonçalves da Silva

Automatic grammar induction for portuguese

Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Thiago Alexandre Salgueiro Pardo

USP – São Carlos
July 2024

Este trabalho é dedicado aos meus pais, Nelson e Miranei, à minha saudosa bisavó, carinhosamente chamada de "mainha", à minha esposa Thays, e a todos que persistem na busca de seus sonhos, enfrentando as adversidades da vida sem desistir.

AGRADECIMENTOS

Apesar de eu não ser uma pessoa muito religiosa, eu preciso agradecer a Deus por ter me ajudado nesta jornada.

Agradeço imensamente ao meu pai, Nelson, pelo apoio incansável e incentivo inabalável que me acompanhou desde os primeiros passos no curso técnico em informática até este momento crucial na USP. Minha mãe, Miranei, merece todo o reconhecimento por sua fé inabalável em mim, mesmo nos momentos em que eu mesmo duvidava. À minha esposa, expresso minha gratidão por sua compreensão indispensável ao longo dessa jornada desafiadora. Lidar com a distância e o escasso tempo juntos durante o mestrado não foi tarefa fácil.

Agradeço à UFCG, onde concluí minha graduação, por fornecer uma base teórica robusta em computação. Ao IFAM, expresso minha gratidão pelo afastamento com vencimento, que me permitiu dedicar-me integralmente ao mestrado. Quero reconhecer profundamente meu orientador, Thiago Pardo, por sua paciência, dedicação, habilidade de comunicação e valiosos *insights* sobre o que eu precisava executar. À biblioteca do ICMC, agradeço por disponibilizar livros não acadêmicos, que foram uma importante válvula de escape nos momentos mais estressantes. Por fim, expresso minha gratidão ao ICMC e à USP pelo acolhimento, possibilitando ao aluno estabelecer um vínculo identitário significativo com a instituição.

Desde a graduação, meu foco sempre foi direcionado ao mestrado no USP-ICMC. Em 2017, submeti minha inscrição no processo seletivo, porém não obtive aprovação. Em contrapartida, fui aceito no mestrado na UFRGS. Devido a alguns equívocos e circunstâncias inevitáveis, não pude concluir o programa na UFRGS. No entanto, longe de desanimar, preparei-me e participei novamente do processo seletivo para o mestrado no USP-ICMC, cinco anos após a primeira tentativa. Na USP optei por dedicar-me integralmente. No entanto, o estudo intensivo apresenta desafios, incluindo fadiga mental, ansiedade e até momentos depressivos. Meu orientador desempenhou um papel crucial na comunicação e orientação, contribuindo para que pudesse enfrentar esses momentos. Almejo retornar em breve para cursar o doutorado nesta renomada instituição.

Este trabalho foi realizado no âmbito do Centro de Inteligência Artificial da Universidade de São Paulo (C4AI - http://c4ai.inova.usp.br/), com o apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (processo FAPESP 2019/07665-4) e da IBM. Este projeto também foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei N. 8.248, de 23 de outubro de 1991, no âmbito do PPI-Softex, coordenado pela Softex e publicado como Residência em TIC 13, DOU 01245.010222/2022-44.

“Encontre um caminho ”

(Diana Nyad)

“Eu era uma pessoa comum que estudava muito. Não existe pessoas milagrosas. ”

(Richard Feynman)

RESUMO

SILVA, D. P. G. **Indução gramatical automática para o português**. 2024. 157 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

A indução gramatical automática é uma tarefa que busca extrair estruturas sintáticas de sentença não anotada. Esta tarefa é importante para diversas aplicações não apenas para Processamento de Língua Natural, mas também para Bioinformática, Linguística, Engenharia de Software e Psicolinguística, para citar algumas. Há uma grande limitação de trabalhos para o português, pois a maioria são direcionados para a língua inglesa. Os trabalhos existentes para outras línguas são construídos, geralmente, para tentar generalizar para outras línguas, que naturalmente podem apresentar estruturas linguísticas distintas. Com a importância da língua portuguesa, uma das 10 mais faladas no planeta, assim como a falta de modelos precisos para a língua portuguesa, faz-se necessário uma investigação sobre a possibilidade de preencher esta lacuna. O objetivo deste trabalho foi estudar os métodos em Indução Gramatical sobre a perspectiva da língua portuguesa e propor novos métodos para o Português usando texto puro (sem nenhum tipo de anotação feita por humanos, ou automatizada não supervisionada). Para atingir estes objetivos, foi realizada uma exaustiva revisão da literatura. Em seguida foram realizados estudos a fim de analisar a viabilidade de determinadas abordagens, como a Informação Mútua, na indução gramatical para o português. Os resultados alcançados neste estudo evidenciam a viabilidade de recuperar estruturas gramaticais, inclusive certos tipos de relações sintáticas, como sujeito e objeto, com uma certa confiança, 74.9% para objetos e 50.1% para sujeitos. Além disso, notou-se que a utilização de características intrínsecas da língua, como o comprimento das palavras, contribuem para um melhor desempenho do método.

Palavras-chave: Indução gramatical, Parsing não supervisionado, Inferência Gramatical.

ABSTRACT

SILVA, D. P. G. **Automatic grammar induction for portuguese**. 2024. 157 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2024.

Grammar induction is a task that aims to extract syntactic structures from unannotated sentences. This task is important for various applications not only in Natural Language Processing but also in Bioinformatics, Linguistics, Software Engineering, and Psycholinguistics, to name a few. There is a significant limitation of works for Portuguese, as most are targeted towards the English language. Existing works for other languages are generally built to generalize to other languages, which naturally may have different linguistic structures. Given the importance of the Portuguese language, one of the top 10 spoken languages ones the planet, as well as the lack of precise models for Portuguese, there is a need for investigation into the possibility of filling this gap. The goal of this work was to study grammar induction methods from the perspective of the Portuguese language and propose new methods for Portuguese using raw text (without any type of annotation made by humans or unsupervised automatization). To achieve these objectives, an exhaustive literature review was conducted. Studies were then carried out to analyze the feasibility of certain approaches, such as Mutual Information, in grammar induction for Portuguese. The results obtained in this study demonstrate the feasibility of recovering grammatical structures, including certain types of syntactic relationships, such as the subject, with a certain level of confidence. Additionally, it was observed that the use of intrinsic language features, such as word length, contributes to improved method performance.

Keywords: Grammar Induction, Unsupervised parsing, Grammatical Inference.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CCG	*Combinatory Categorical Grammar*
CCM	*Constituent Context Model*
DAA	*Directed Attachment Accuracy*
DDA	*Direct Dependency Accuracy*
DMV	*Dependency Model with Valence*
EM	*Expectation-Maximazation*
FNC	Forma Normal de Chomsky
GLC	Gramática Livre de Contexto
HMM	*Hidden Markov Models*
IG	Indução Gramatical
IGNS	Indução Gramatical Não Supervisionada
IGS	Indução Gramatical Supervisionado
IGSS	Indução Gramatical Semi-Supervisionada
IM	Informação Mútua
IO	*Inside Outside*
KL	*Kullback-Leibler*
LTGM	*Latent Tree Graphical Model*
MCMCS	*Chain Monte Carlo Sampling*
ML	*Maximum Likelihood*
MLE	*Maximum Likelihood Estimation*
PLN	Processamento de Línguas Naturais
PUD	*Parallel UD*
SN	Sintagmas Nominais
SV	Sintagmas Verbais
UAA	*Undirected Attachment Accuracy*
UAS	*Unlabeled Attachment Score*
UD	*Universal Dependencies*
UDA	*Undirected Dependency Accuracy*
UDOP	*Unsupervised Data-Oriented language Processing*
VI	*Variational Inference*
WSJ	*Wall Street Journal*

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

1.1 Contextualização e Motivação

O Processamento de Línguas Naturais (PLN) é uma área de pesquisa que busca aplicar técnicas computacionais em máquinas para aprender, compreender e produzir a língua humana (HIRSCHBERG; MANNING, 2015). Nos últimos anos, o PLN tem causado grande impacto na sociedade através do desenvolvimento de aplicações como sistemas de tradução, sistemas de pergunta e resposta, e reconhecimento de fala. Mais recentemente, essas aplicações alcançaram avanços com o uso de *large language models*, modelos de língua treinados em uma grande quantidade de dados como o GPT3 (STOKEL-WALKER; NOORDEN, 2023; LASKAR *et al.*, 2023).

Sistemas de tradução são uma das aplicações de PLN mais antigas. Após décadas de desenvolvimento, houve grandes avanços, mas ainda há muito trabalho a ser feito devido às diferenças sintáticas e léxicas existentes nos diferentes idiomas (WANG *et al.*, 2021), assim como a ambiguidade natural da língua humana, e às línguas com poucos recursos (línguas indígenas, por exemplo) (STAP; ARAABI, 2023). Por exemplo, a sentença *"Eu tenho saudade de você"* traduzida para a língua inglesa equivale à *"I miss you"*. A tradução desta sentença não pode ser realizada palavra por palavra. É preciso compreender como as sentenças são construídas e as relações entre as palavras da sentença em ambas as línguas. A área da Linguística que estuda as estruturas das sentenças e suas relações denomina-se sintaxe. Essa área é definida como o estudo da organização das palavras (em termos de ordenação e estruturação) na formação de sentenças. Esse entendimento é compartilhado por diferentes visões de teorias sintáticas (KULMIZEV; NIVRE, 2022).

Quase toda aplicação de PLN necessita de algum conhecimento sintático para obter bons resultados. Sistemas de simplificação de textos, revisores gramaticais e sistemas de extração de informação são algumas das aplicações que se beneficiam da representação explícita da sintaxe

(SIDDHARTHAN, 2014). Em aplicações que utilizam modelos de língua, a sintaxe pode ser aprendida mesmo que ela não tenha sido explicitada, a depender do tipo de tarefa (KULMIZEV; NIVRE, 2022) (JAWAHAR; SAGOT; SEDDAH, 2019)

Aplicações de PLN, geralmente, obtêm conhecimento sintático a partir do processamento de córpus anotado. Córpus pode ser descrito como um grande conjunto de dados linguísticos que evidenciam o uso da língua obedecendo determinados critérios como formatação, tipos de texto e propósito (MCENERY, 2019). As especificações sintáticas descritas para cada sentença no córpus são denominadas de anotação (SARDINHA, 2004). Realizar a anotação manual de um corpus é um processo dispendioso em termos financeiros e consome considerável tempo. Para agilizar esse procedimento, recorre-se à anotação automática (HOVY; LAVID, 2010). Essa tarefa é aprendida a partir de córpus já anotado por humanos, e aplica esse aprendizado em texto sem nenhuma anotação. No entanto, para línguas com pouca ou sem nenhuma anotação, a utilização dessa abordagem não é viável porque é necessário um número mínimo de amostras para que seja possível aprender os padrões.

Uma alternativa que possibilita a extração de estruturas sintáticas de um córpus não anotado é a utilização de Indução Gramatical (IG) . Várias áreas do conhecimento são beneficiadas pela IG (HIGUERA, 2005). Na área da visão computacional, ela é empregada para extrair informações de imagens (SHI *et al.*, 2019), bem como para aplicar técnicas de processamento de imagens e visão computacional na indução de gramática (LI *et al.*, 2022). Na engenharia de Software, há um grande interesse em geração de código a partir do uso de IG (STEVENSON; CORDY, 2014), inclusive para linguagens de domínio específico. Em compiladores e em teoria de linguagens formais, IG é bastante utilizada para inferir gramaticas regulares e sensíveis ao contexto (SHIBATA, 2021) (COHEN *et al.*, 2017). Em bioinformática, IG é utilizada para inferir estruturas de DNA desconhecidas ou difíceis de serem encontradas em grandes bases de dados, como a proteína Amiloide (DYRKA *et al.*, 2021) (UNOLD; GABOR; DYRKA, 2020). A busca de proteínas em sequências de DNA é útil na produção de novos medicamentos (BATISTA *et al.*, 2012). Em psicolinguística, IG é utilizada para construir modelos de aquisição da linguagem (BOD, 2009) (WINTNER, 2010) (BANNARD; LIEVEN; TOMASELLO, 2009) (CLARK, 2004). Na Linguística, pode ser útil para aprender a gramática de línguas mortas ou línguas com escassez de recursos (como as línguas indígenas) (DAHL *et al.*, 2023).

Tradicionalmente, em PLN, IG é definida como uma tarefa não supervisionada, que tem como objetivo inferir a estrutura sintática de sentenças sem uso de anotações sintáticas (KLEIN; MANNING, 2004). No entanto, é possível encontrar estudos que utilizam o termo "indução gramatical" para se referirem a tarefas que utilizam anotações sintáticas (tarefas supervisionadas). Por convenção, neste trabalho, utilizaremos os termos Indução Gramatical Semi-Supervisionada (IGSS) para designar trabalhos que utilizam parcialmente dados anotados, Indução Gramatical Supervisionado (IGS) para representar os trabalhos que utilizam apenas dados anotados, Indução Gramatical Não Supervisionada (IGNS) para designar os trabalhos que não utilizam dados

anotados e IG para uso genérico.

IG, por ser utilizada em várias áreas do conhecimento, acaba por ser substituída por outros termos como: aquisição de gramática *("grammar acquisition")* (JIN *et al.*, 2018), aprendizado de gramática não supervisionada *("unsupervised grammar learning")* (COHEN; SMITH, 2009), *parsing* não supervisionado *("unsupervised parsing")* (KLEIN; MANNING, 2002), indução de árvore latente *("latent tree induction")* (ANDREW, 2019) e inferência gramatical *("grammatical inference")* (HIGUERA, 2005). Os termos *"grammar induction"* e *"unsupervised parsing"* são os mais utilizados na comunidade de PLN.

A IGNS trabalha com duas representações de gramática: gramática de constituintes e gramática de dependência (JURAFSKY, 2000). A primeira estuda como as sentenças são constituídas por blocos básicos. Por exemplo, a expressão *"O menino chutou a bola"*, retratada na árvore sintática de constituintes apresentada na Figura 1, é composta por dois artigos, dois substantivos e um verbo. Estes constituintes se organizam em Sintagmas Nominais (SN) e Sintagmas Verbais (SV), que, por sua vez, se agrupam para constituir a sentença (S).

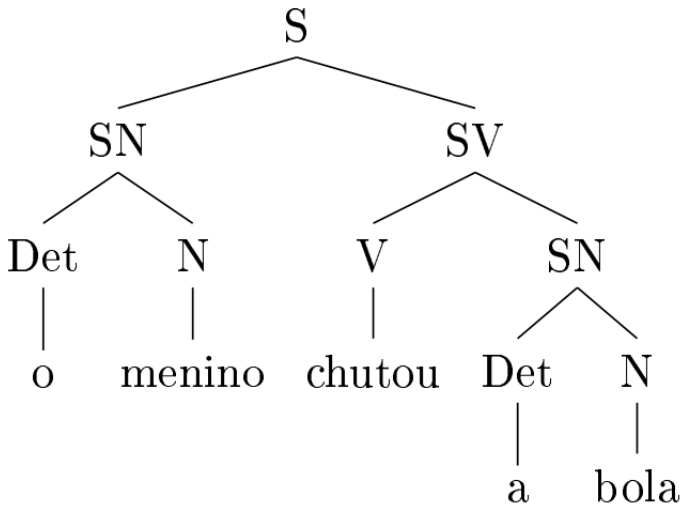


Figura 1 – Árvore sintática com representação de constituintes. SN, Det, N, V e SN são constituintes. As palavras são as folhas da árvore.

Fonte: (PAGANI, 2018, pag.12)

Árvore sintática referente à gramática de constituintes pode ser representada, e gerada, usando Gramática Livre de Contexto (GLC) conforme apresentado na Figura 2. Cada linha representa uma regra de produção. O termo do lado esquerdo gera os termos do lado direito. Por exemplo, na primeira linha da Figura 2, o termo S gera os termos SN e SV da Figura 1. GLC é explicada em mais detalhes na Seção 2.1.1.

No modelo de dependência, estabelecem-se relações de dependências diretamente entre as palavras. A mesma sentença, representada na Figura 1, apresenta relações de sujeito (nsubj) (entre o verbo "chutou" e a palavra "menino") e objeto (obj) (entre o verbo e a palavra "bola"), por exemplo.

S → SN SV
SN → Det N
SV → V SN
Det → o | a
N → menino | bola
V → chutou

Figura 2 – Gramática livre de contexto usada para representar a árvore de constituintes da Figura 1

Fonte: Próprio autor

Enquanto a gramática de constituintes é baseada na constituição e construção das sentenças, a gramática de dependência, além de conceder importância para a construção das sentenças, também considera as funções que as relações entre as palavras exercem. A mesma sentença, representada na Figura 1, é representada na Figura 3 como árvore de dependência. Nessa árvore de dependência, busca-se representar as relações existentes entre as palavras da sentença (MARNEFFE; NIVRE, 2019). Por exemplo, a relação de sujeito (nsubj) está representada por um arco que conecta as palavras "chutar" (verbo) e "menino" (substantivo). O primeiro é chamado de termo "cabeça" e o segundo de termo "dependente". A gramática de dependência pode ser representada usando o formalismo proposto por Hays (1964). Segundo esse formalismo, a gramática de dependência pode ser descrita com base no uso de regras que podem ser apresentadas de duas formas diferentes (COURTIN; GENTHIAL, 1998):

1. * (X)
2. $X_1 ... X_i * X_{j+1} ... X_n$

sendo a variável X uma categoria morfossintática. A primeira regra define a raiz da árvore, e a segunda regra os ramos da árvore. O asterisco (*) representa a localização do pai do ramo da árvore que está sendo construída, ou seja, o termo cabeça da relação. É possível gerar a frase "o menino chutou a bola" usando diferentes conjuntos de regras. Uma forma possível são as seguintes regras:

- ***(verbo)** → root[1] (chutou)
- **verbo (substantivo,*,substantivo)**→ chutou (menino,*,bola)
- **substantivo (determinante,*)** → menino (o,*) | bola (a,*)

Essas regras geram as relações de dependência raiz(chutou), <chutou,menino>, <bola,chutou>, <a,bola>, <o,menino> que são apresentadas na Figura 3. O formalismo definido por Hays (1964)

[1] Raiz da árvore de dependência

pode ser usado para gerar diferentes sentenças usando as mesmas categorias morfossintáticas, por exemplo: “a bola chutou o menino”[2]. A intenção geral deste trabalho foi induzir a gramática, mesmo que parcialmente, a partir da descoberta destas regras.

A árvore apresentada na Figura 3 utiliza o *framework Universal Dependencies* (UD) (MARNEFFE *et al.*, 2021). A UD têm uma missão ousada: buscar representar diferentes línguas em gramática de dependência usando um mesmo grupo de categorias morfossintáticas e relações sintáticas. A UD é descrita em detalhes na Subseção 2.5.

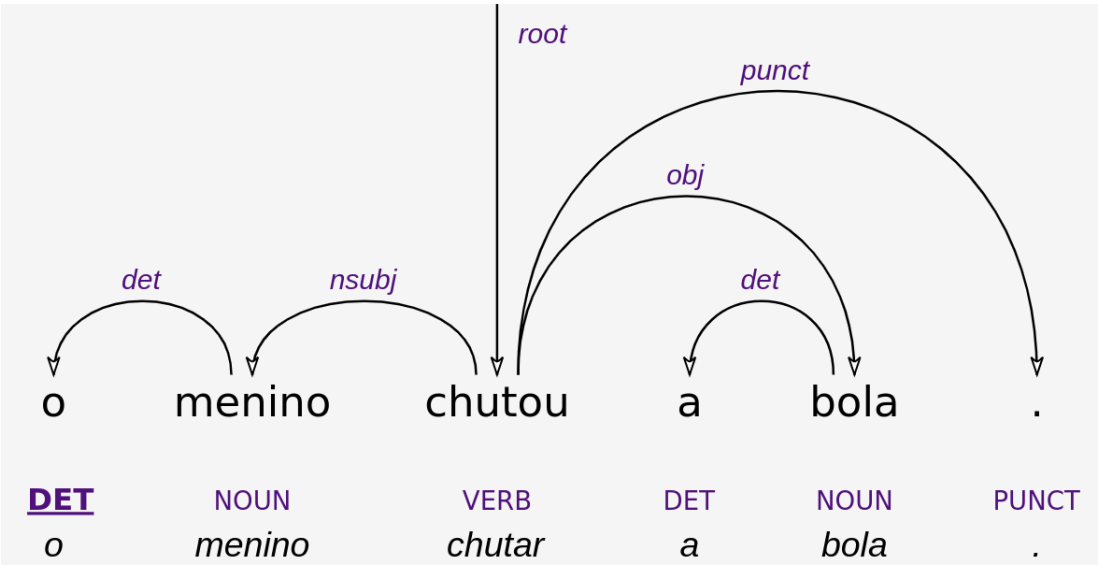


Figura 3 – Árvore sintática com representação de dependência.

Fonte: Próprio autor com suporte da ferramenta Arborator-Grew (GUIBON *et al.*, 2020)

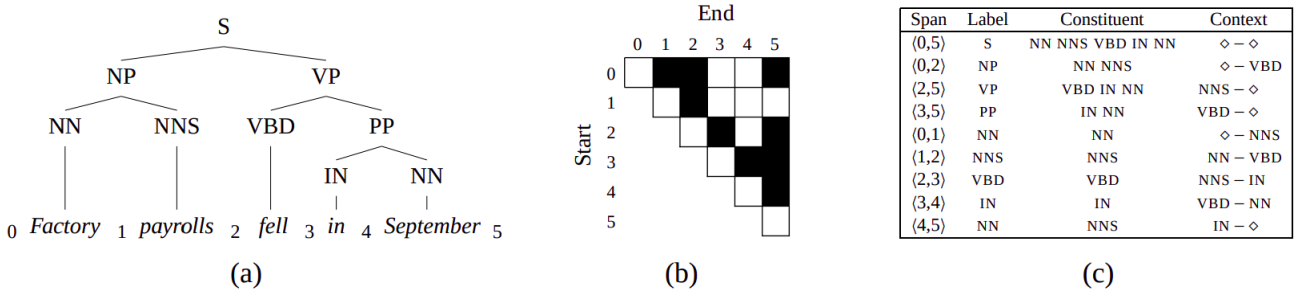
A tarefa de extrair estruturas sintáticas de um texto sem nenhum tipo de anotação já foi considerada como uma missão impossível (GOLD, 1967).

A maioria das abordagens de IGNS das duas últimas décadas são gerativas, principalmente em gramática de dependência (GRAVE; ELHADAD, 2015). Os modelos gerativos *Constituent Context Model* (CCM)(KLEIN; MANNING, 2002), para gramática de constituintes, e *Dependency Model with Valence* (DMV)(KLEIN; MANNING, 2004), para gramática de dependência, exerceram grande influência na implementação de novas abordagens.

O modelo CCM constrói a árvore sintática de forma ascendente (dos terminais até a raiz). A ideia basicamente é estimar onde começa e termina cada sintagma da sentença. Essa separação é feita com uso de colchetes. Por exemplo, a sentença apresentada na Figura 1 pode ser representada usando colchetes da seguinte forma *[[o menino][chutou[a bola]]]*. Para estimar qual a melhor organização dos colchetes para determinada sentença, Klein e Manning (2002) utilizaram o algoritmo *Expectation-Maximazation* (EM). Um exemplo de indução realizada pelo CCM é apresentado na Figura 4.

[2] Apesar de a frase ser gramaticamente correta, ela é semanticamente incorreta

A Figura 4(A) apresenta um exemplo de *parsing* em árvore de constituintes. Essa árvore pode ser representada usando matriz de *parsing*, conforme apresentada na Figura 4(B). A identificação de cada constituinte na árvore de constituinte e os sintagmas que são formados é apresentada na Figura 4(C). Por exemplo, a palavra "Factory" é representada em (A) delimitada pelos espaços 0 e 1 e na matriz (B) identificada pela linha 0 e coluna 1. Na Figura 4(C), "Factory" é representada na linha 5 com sua respectiva classe morfossintática (NN), os constituintes representados no intervalo (NN) e o contexto (◇–NNS), onde ◇ representa o limite da sentença.



Span	Label	Constituent	Context
⟨0,5⟩	S	NN NNS VBD IN NN	◇ – ◇
⟨0,2⟩	NP	NN NNS	◇ – VBD
⟨2,5⟩	VP	VBD IN NN	NNS – ◇
⟨3,5⟩	PP	IN NN	VBD – ◇
⟨0,1⟩	NN	NN	◇ – NNS
⟨1,2⟩	NNS	NNS	NN – VBD
⟨2,3⟩	VBD	VBD	NNS – IN
⟨3,4⟩	IN	IN	VBD – NN
⟨4,5⟩	NN	NNS	IN – ◇

Figura 4 – Exemplo de indução

Fonte: (KLEIN; MANNING, 2002, Pag. 2)

O modelo DMV constrói a árvore de forma descendente (da raiz até os terminais). Neste modelo, cada nó da árvore sintática é gerado recursivamente. Por exemplo, na sentença representada na Figura 3, a raiz "chutou" é gerada no início do modelo. Em seguida, "chutou" gera "menino" à esquerda, que gera "o", também à esquerda. Uma vez que as palavras "o" e "menino" não têm mais nada a ser gerado, o processo continua do lado direito. Para gerar a árvore, é utilizada a distribuição de probabilidade $P(\neg STOP|h,dir,a)$, onde, dados h, dir e a, que representam o termo cabeça da relação, a direção de geração e o termo dependente da relação, respectivamente, é estimada a probabilidade de a palavra ser gerada. Para estimar os parâmetros da distribuição é calculada a *Maximum Likelihood Estimation* (MLE) a partir do uso dos algoritmos EM e *Inside Outside* (IO)[3].

Os modelos discriminativos ainda são pouco explorados para tarefas de IGNS, mas nos últimos anos têm ganhado espaço. Grave e Elhadad (2015), a fim de induzir estruturas de dependência não projetivas, ou seja, arcos de dependência que podem se cruzar, utilizam vetores multidimensionais, para representar as características do modelo, e otimização convexa. Além disso, induzir estruturas de dependência não projetivas é bastante difícil em algumas línguas como tcheco e holandês (GRAVE; ELHADAD, 2015). Com advento de redes neurais, alguns estudos aplicam *autoencoders* para induzir gramática de dependência (CAI; JIANG; TU, 2017). Há também modelos que utilizam as duas abordagens (LI *et al.*, 2019) (HAN; JIANG; TU, 2019).

[3] Em linhas gerais, tratam-se de algoritmos utilizados para estimar variáveis não observáveis, no caso de IG, as árvores. Esses algoritmos são explicados em detalhes na Seção 2.3.1.2

Apesar de um dos primeiros trabalhos que utilizou informação mútua para realizar *parsing* de língua ter sido proposto antes dos primeiros modelos de IG (MAGERMAN; MARCUS, 1990), apenas recentemente que Informação Mútua (IM) começou a ganhar espaços em modelos de IG. IM mede o nível de associação entre duas variáveis aleatórias. Em PLN, a IM mede o nível de informação em comum entre duas palavras, ou seja, o quanto elas estão associadas uma a outra (CHURCH; HANKS, 1990). Por exemplo, "maçã" e "banana" apresentam informação mútua maior que "peixe" e "bicicleta". Recentemente, Futrell *et al.* (2019) constataram que pares de palavras que têm relação sintática apresentam maior informação mútua que pares de palavras que não têm relação. Usando dessa ideia, Hoover *et al.* (2021) aplicam informação mútua para IGNS de dependência utilizando modelos de linguagem.

Os atuais métodos do estado-da-arte utilizam abordagem neurais, tanto para gramática de dependência quanto para gramática de constituintes. Yang *et al.* (2020) atingiram o estado-da-arte ao construírem o modelo probabilístico com mais de um nível de distância de hierarquia entre os nós na árvore. Outros trabalhos atingiram o estado-da-arte ao estender o modelo DMV para redes neurais (HAN; JIANG; TU, 2019) (HAN; JIANG; TU, 2017) (JIANG; HAN; TU, 2016). (SHEN *et al.*, 2021) usa o conceito de distância e altura sintática para segmentar a sentença em partes menores. Todos estes trabalhos utilizam algum tipo de informação léxica para contribuir com o desempenho. Recentemente, Andrew (2019) utilizaram redes neurais aplicadas ao algoritmo IO usando texto puro. Shen *et al.* (2021) utilizaram mecanismos de atenção (o modelo "tem mais atenção" para algumas palavras do que para outras) para induzir gramática de dependência e constituinte de texto puro.

A tarefa de IGNS apresenta muitos desafios, especialmente para as línguas menos favorecidas em dados. Este trabalho busca analisar diferentes abordagens existentes para a língua portuguesa a fim de avaliar a criação e aplicabilidade de novos métodos.

1.2 Objetivos e hipóteses de pesquisa

Trabalhos de IGNS para a língua portuguesa são bastante limitados. O nosso objetivo nesta dissertação foi desenvolver, e avaliar métodos especificamente para o português com a finalidade de obter melhor desempenho que os métodos atuais. Esse trabalho considerou as seguintes hipóteses:

1. Há características específicas da língua portuguesa que contribuem para um melhor desempenho em tarefas de IGNS.

2. É possível desenvolver técnicas melhores de IGNS para a língua portuguesa que os sistemas generalistas atuais.

Para alcançar os objetivos deste trabalho, foram investigados métodos clássicos e mais

modernos para o português. Dois métodos foram escolhidos para reprodução: DMV (KLEIN; MANNING, 2004) e (HE; NEUBIG; BERG-KIRKPATRICK, 2018), que aplica redes neurais com o DMV. Para a execução destes métodos, utilizaremos os córpus disponibilizados pela UD (MARNEFFE *et al.*, 2021) em língua portuguesa. Para o desenvolvimento de um novo método, o uso de informação mútua foi explorado.

Com este trabalho, espera-se contribuir para potencializar a criação de recursos, ferramentas e aplicações para a língua portuguesa. Essa contribuição permitirá reduzir a lacuna da língua portuguesa em tarefas de PLN em comparação com línguas com mais recursos como as línguas inglesa, chinesa e espanhola.

1.3 Organização do texto

O Capítulo 2 apresenta a fundamentação teórica relacionada com este trabalho de pesquisa, como conceitos e tipos de gramática, características de gramática de constituintes e gramática de dependência, as diferentes abordagens utilizadas na tarefa de IGNS, os diferentes tipos de córpus e suas características e, por fim, avaliação de modelos. Capítulo 3 descreve um mapeamento completo da área nos últimos 20 anos, apresentando cinco questões de pesquisas que são respondidas. Em seguida, o Capítulo 4 apresenta a aplicação do uso de IM na descoberta de relações de dependência. Este estudo foi reproduzido a partir do estudo de Futrell *et al.* (2019) e demonstrou que alguns tipos de relações sintáticas apresentam maior IM que outras, a exemplo da relação sintática de sujeito e objeto . O Capítulo 5, é oferecida uma análise mais abrangente em 69 idiomas sobre o impacto da IM na identificação de relações sintáticas, assim como as similaridades e disparidades entre esses idiomas. O Capítulo 6 apresenta a eficiência do uso da IM para indução gramatical em línguas indígenas, notoriamente línguas com poucos recursos. O Capitulo 7 apresenta uma avaliação experimental em indução gramatical com diferentes métodos. O estudo busca responder as hipóteses apresentadas. Finalmente, no Capítulo 8 é apresentada a conclusão do estudo.

CAPÍTULO

2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos necessários para a compreensão deste trabalho. Na Seção 2.1, apresentamos conceitos sobre gramática e suas formas de representação para uso em sistemas computacionais. Na Seção 2.2, descrevemos o que é indução gramatical. Na Seção 2.3, apresentamos os principais métodos utilizados na tarefa de indução gramatical, assim como uma breve explanação matemática necessária para o entendimento dos métodos. Na Seção 2.4, apresentamos conceitos sobre córpus. Na Seção 2.5, apresentamos o modelo *Universal Dependencies*, utilizado neste trabalho de mestrado. Na Seção 2.6, apresentamos as principais medidas de avaliação da tarefa.

2.1 Gramática

A palavra gramática é originada da expressão grega $\gamma\rho\alpha\mu\mu\alpha\tau\iota\kappa\eta\ \tau\varepsilon\chi\upsilon\eta$ que significa "arte gramatical" (MATTHAIOS, 2011). Essa expressão era usada para se referir à disciplina voltada para a arte de escrever literatura. Segundo Aulete (1881, pag. 1974), gramática é definida como a "*ciência das leis que regem a formação e estrutura das línguas*". Adicionalmente, a gramática pode ser conceituada como a teoria da linguagem, isto é, como a língua é estruturada, gerada e transmitida. A gramática gerativa (CHOMSKY, 2014) e a gramática cognitiva (LANGACKER, 1987), são as duas mais proeminentes teorias sobre a linguagem (HARRIS, 2021). Essas também são consideradas como abordagens racionalista e empirista, respectivamente (DALE; MOISL; SOMERS, 2000).

Na linha racionalista, Chomsky defende a ideia de que um conjunto infinito de sentenças pode ser **gerado** a partir de um conjunto de regras, que operam sobre um conjunto de símbolos, que são combinados, rearranjados e transformados (CHOMSKY, 1956). Ainda segundo Chomsky, a linguagem compreende duas grandes habilidades: competência e performance. Competência refere-se ao conhecimento do falante e do ouvinte sobre a língua. Performance refere-se à capacidade de o falante e o ouvinte em usar a língua, seja para gerá-la ou compreendê-la

(CHOMSKY, 2014). Na visão de Chomsky, tanto a performance quanto a competência são inatas, uma vez que ele acredita que há uma pré-definição do conhecimento linguístico quando a criança nasce, o que ele chama de *"language acquisition device"*(dispositivo da aquisição da linguagem) (CHOMSKY, 2002). Diferenças intelectuais e distúrbios de linguagens, com origem genética ou gestacional, são geralmente usadas como argumentos em favor da teoria gerativa (SCIULLO *et al.*, 2010). Apesar de a performance linguística ser aceita na academia como algo inato, a competência inata é rejeitada pelos cognitivistas (TOMASELLO, 2000).

O desacordo em considerar a competência linguística como inata contribuiu para o surgimento da gramática cognitiva (LANGACKER, 1987) e da linguística cognitiva (LAKOFF, 1987). Ambas surgiram, oficialmente, no mesmo ano, mas apresentam definições distintas. Enquanto a primeira refere-se a uma teoria da linguagem, a segunda refere-se a um programa de pesquisa onde a gramática cognitiva está incluída (MITKOV, 2022). Diferentemente da gramática gerativa, a gramática cognitiva despreza o conceito de regras da gramática definidas por Chomsky e formalizada com a GLC. Para LAKOFF (1987), a língua é constituída por construções, que não são geradas por regras pré-determinadas, mas sim com base no uso, isto é, de forma empírica. Segundo essa visão, a criança, que está aprendendo a primeira língua, consegue utilizar corretamente construções sintáticas não porque ela usa regras de um "dispositivo da aquisição da linguagem", mas porque ela aprendeu como usá-las com base no uso (passivo ou ativo). Chomsky, por outro lado, não concorda com essa visão, pois, segundo ele, a criança não recebe estímulos suficientes para compreender as regras sintáticas (BERWICK *et al.*, 2011).

Para os teóricos, tanto os gerativos quanto os cognitivistas, a gramática é composta principalmente por fonologia, sintaxe e semântica (CHOMSKY, 2014; EVANS, 2006). No entanto, a gramática gerativa é baseada quase no seu todo na sintaxe, enquanto a gramática cognitiva valoriza a importância da semântica na estruturação das sentenças (HARRIS, 2021). Para os linguistas mais aplicados, a fonologia, a morfologia e a sintaxe representam componentes da gramática (AKMAJIAN *et al.*, 2017). Outros, além da fonologia, da morfologia e da sintaxe, incluem também a semântica (BECHARA, 2012). Na computação, especificamente no campo da teoria da computação e linguagens de programação, considera-se como gramática apenas a morfologia, a semântica e, principalmente, a sintaxe (AHO; SETHI; ULLMAN, 2007).

2.1.1 Formas de representação de gramática

Na década de 50, Chomsky propôs um formalismo da língua que viria a exercer nas décadas posteriores forte influência sobre as áreas de linguística e computação (HARRIS, 2021). Nesse formalismo, Chomsky (1956) descreveu uma hierarquia de gramáticas fundamentada em um conjunto de regras, que definia quais conjuntos de símbolos a gramática era capaz de gerar. Apesar de esse formalismo não ser muito aceito por linguistas e por ser um problema em aberto sobre em qual hierarquia a linguagem natural se encontra (JÄGER; ROGERS, 2012), o uso deste formalismo é bastante útil para a computação. Entre as gramáticas descritas na

hierarquia de Chomsky, a GLC, que também foi definida por JW (1959) de forma independente, contribuiu com a construção das primeiras linguagens de programação (BACKUS *et al.*, 1960) e no desenvolvimento dos primeiros sistemas de PLN (JURAFSKY, 2000).

Há outros tipos de gramática que são utilizados em alguns trabalhos de IG como a *Combinatory Categorical Grammar* (CCG) (STEEDMAN; BALDRIDGE, 2011), que buscou mapear sons e significados da língua. Outros dois formalismos bastante importantes da computação, e os mais utilizados em tarefas de IG, são a gramática de constituintes e a gramática de dependência.

2.1.1.1 gramática de constituintes

gramática de constituintes ou gramática de estrutura de sentenças (*phrase structure grammar*) foi formalizada por Chomsky usando GLC (CHOMSKY, 1956), conforme o descreve em *Aspects of the Theory of Syntax*:

> *"An unordered set of rewriting rules, applied in the manner described loosely here (and precisely elsewhere), is called a constituent structure grammar (or phrase structure grammar). The grammar is, furthermore, called context-free..."* (CHOMSKY, 2014, pag.71)

Toda GLC pode gerar um conjunto infinito de *strings* (uma ou mais combinações de símbolos), dentro de um domínio definido. Por exemplo, a GLC apresentada na Figura 5, pode gerar um número infinito de sentenças de diferentes tamanhos apenas com a expressão *gata e dorme e gato acorda* de forma repetitiva. Os símbolos que compõem cada *string* representam o alfabeto utilizado na gramática. Estes símbolos, formalmente, são chamados de terminais em GLC. Pode-se observar um exemplo na Figura 5 em que o conjunto dos terminais é composto por $\{$*dorme, acorda, gata, gato, e, não*$\}$.

Apesar de GLC permitir gerar infinitas sentenças, há limites computacionais. É impossível gerar todas as sentenças possíveis de um conjunto S usando uma única GLC (SIPSER, 2021). Toda GLC precisa ter um domínio definido de quais *strings* podem ser geradas. O domínio é definido pelas regras de produção. São elas que determinam o que pode ser gerado pela GLC. Cada linha na Figura 5(a) representa uma regra de produção. As regras que são produzidas pelos mesmos não terminais podem ser aglutinadas na mesma linha, conforme apresentado na Figura 5(b).

A regra $C \rightarrow e$ significa que a partir da variável **C** é possível gerar o terminal **e**. Assim como, para gerar os terminais **gata dorme** a partir do não terminal **S**, é preciso gerar os não terminais **N X**. Finalmente, os não terminais **N X** geram respectivamente **gata dorme**. Esse conjunto de gerações pode ser representado como uma árvore, apresentada na Figura 6.

A gramática livre de contexto é definida como uma quadrupla **(V, Σ, R, S)** onde;

```
S → N X                    S → N X
X → X C V                  X → X C V | V
X → V                      V → dorme | acorda
V → dorme                  N → gato  | gata
V → acorda                 C → e     | e T
N → gato                   T → não
N → gata
C → e
C → e T
T → não
(a)                              (b)
```

Figura 5 – GLC que permite geração de sentenças infinitas

Fonte: Próprio autor

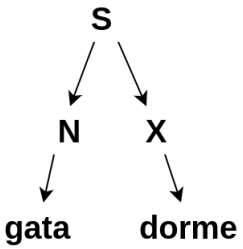


Figura 6 – Exemplo de geração usando a GLC da Figura 5

Fonte: Próprio autor

1. **V** representa as variáveis
2. Σ representa os terminais
3. **R** representa as regras de produção.
4. **S** Representa a variável inicial.

A GLC também permite a produção de um termo vazio, representado pelo símbolo ε. Esse símbolo é utilizado para interromper a geração de uma sequência de símbolos. Por exemplo, na GLC descrita na Figura 5, para que a gramática gere apenas a palavra **gato**, é necessário adicionar uma nova regra [X $\rightarrow \varepsilon$]. O não terminal responsável por iniciar a produção da árvore sintática é chamado de variável inicial, representada por **S**. A GLC apresentada na Figura 5 é descrita na forma de quadrupla **(V, Σ, R, S)** como: ({ X, V, N, C, T},{ e, não, gata, gato, dorme, acorda}, regras de produção da Figura 5(b), S).

A gramática de constituintes está inteiramente conectada com a gramática gerativa, pois apresenta a mesma proposta central: gerar sentenças da língua a partir do uso de regras. Apesar de que, diferente da língua natural, não é possível gerar todas as sentenças possíveis da língua natural usando uma GLC. Assim como a língua natural, as GLC também são ambíguas. No entanto, é possível, para alguns casos, transformar uma GLC ambígua para não ambígua usando

a Forma Normal de Chomsky (FNC) (CHOMSKY, 1959). O objetivo principal da FNC não é excluir a ambiguidade, pois são poucos os casos em que isso é possível, mas torná-la mais simples, o que facilita a compreensão e reduz a complexidade computacional (SIPSER, 2021). Eventualmente, é permitido o uso da produção $S \rightarrow \varepsilon$. A FNC não permite derivação à esquerda, o que é particularmente útil para a tarefa de IG em GLC. Qualquer GLC pode ser definida em FNC, assim como toda FNC pode ser definida em GLC. Os detalhes da transformação de GLC em FNC podem ser encontrados em Sipser (2021, Pag. 102).

Apesar de a GLC ser concebida para gramática de constituintes, a geração da árvore sintática de constituintes não está restrita ao uso de GLC. Outros formalismos também podem ser empregados, como CCG (STEEDMAN; BALDRIDGE, 2011), DCG (*Definite-Clause Grammar*) (EL-SHISHINY, 1990), e LFG (*Lexical Functional Grammar*) (BRESNAN *et al.*, 2015). Neste trabalho, nos concentramos exclusivamente na GLC para a geração de árvores de constituintes.

Na gramática de constituintes, a árvore sintática pode ser construída de duas formas: cima para baixo (*top-down*) ou de baixo para cima (*botton-up*). Apresentamos, na Figura 6, a construção *top-down* para gerar a sentença *gata dorme*. O processo é inverso ao *top-down*, na construção *botton-up*.

A árvore sintática, apresentada na Figura 6, construída de forma *botton-up* inicia-se pelos terminais **gata** e **dorme**. Em seguida, para encontrar os pais de cada terminal, busca-se nas regras de produção os não terminais que geram os terminais **gata** e **dorme**, que são, respectivamente, **N** e **X**. Este processo é executado de forma recursiva até que o terminal raiz (ou variável inicial) é alcançada.

A árvore sintática, como mostrada na Figura 6, é construída de forma *bottom-up*, começando pelos terminais **gata** e **dorme**. Em seguida, para encontrar os pais de cada terminal, consultam-se as regras de produção a fim de encontrar os não-terminais que geram os terminais **gata** e **dorme**, que são, respectivamente, **N** e **X**. Esse processo é executado de forma recursiva até que se alcance o não-terminal raiz (ou variável inicial), **S**. Caso a gramática não esteja na FNC e não permita ambiguidade, é possível gerar diversas variações de árvores sintáticas, sejam elas *botton-up* ou *top-down*.

2.1.1.2 Gramática de Dependência

Antes mesmo de Chomsky publicar a sua famosa obra *Syntatic Structure* (CHOMSKY, 1957) e uma onda de críticas sobre a sua teoria surgir (HARRIS, 2021), Tesnière já estava trabalhando em um novo formalismo para a sintaxe, que não viu ser publicado antes de sua morte (TESNIÈRE, 1959). É dado a Tesnièrre os créditos pela primeira tentativa de formalizar a gramática de dependência, uma vez que esta ainda não está completamente formalizada (MARNEFFE; NIVRE, 2019). Apesar de Tesnière não ter proposto uma teoria de gramática cognitiva, muitos elementos de seu formalismo são semelhantes à gramática cognitiva proposta por Langacker (LANGACKER, 1995).

A gramática de dependência difere da gramática de constituintes em vários pontos. Primeiro, não existe relação terciária em árvore de dependência, diferentemente da árvore de constituintes, em que mais de dois constituintes podem ser filhos de um mesmo constituinte (MARNEFFE; NIVRE, 2019). Segundo, a gramática de dependência valoriza as funções sintáticas existente entre as relações, o que lhe confere uma maior riqueza de informação (MARNEFFE *et al.*, 2021). Terceiro, árvores de constituintes podem crescer exponencialmente no número de nós, diferentemente de árvores de dependência (CANCHO; SOLÉ; KÖHLER, 2004) (SPITKOVSKY; ALSHAWI; JURAFSKY, 2010). Quarto, construir árvores sintáticas a partir de gramática de constituintes de forma ótima é NP-completo (SIMA'AN, 1996), diferentemente da árvore de dependência que pode ser polinomial caso seja projetiva (permite intersecção entre os arcos de dependência) (KUHLMANN, 2010). Por último, a gramática de dependência permite inserir aspectos da gramática de constituintes. Por exemplo, terminais podem ser representados na árvore de dependência como as palavras que não têm dependente, enquanto que os não terminais podem ser representados pelas as palavras que têm dependentes.

Há diferentes formas de representar gramática de dependência. Entre as mais conhecidos estão o *Word Grammar* (MITKOV, 2022, pag.526), *Meaning–Text Theory* (MEL'CUK *et al.*, 1988) e a UD, descrita na Subseção 2.5. Em toda representação de gramática de dependência, um *token* (qualquer representação simbólica indivisível em um texto) pode ser dependente de apenas um *token*. No entanto, o mesmo *token* pode exercer dependência sobre mais de um *token*. Nas representações mais atuais apresentadas nas Figuras 8 e 9, todos os *tokens* devem ser dependente de outro *token*, exceto o *ROOT*.

De acordo com o nosso conhecimento, o primeiro trabalho de IG para gramática de dependência foi proposto por Schank e Tesler (1969). No entanto, esse se trata de um modelo que sofre bastante influência de Chomsky, fugindo assim do formalismo proposto por Tesnière. O trabalho de Carroll e Charniak (1992), é considerado o primeiro trabalho relevante que utilizou gramática de dependência (KLEIN; MANNING, 2004), apesar de terem utilizado uma representação informal. Essa representação é apresentada na Figura 7. As setas saem dos *tokens* dependentes e incidem nos *tokens* que exercem a dependência. Por exemplo, a palavra *"ate"* depende da palavra *"She"*. No entanto, este formalismo não apresenta consistência sobre a direção dos arcos. Alguns trabalhos consideram que o arco incidente deve ser cabeça da relação, outros consideram que deve ser dependente (JURAFSKY, 2000). Neste trabalho, adotamos a direção proposta no *framework* UD, que define a seta do arco como incidente sobre o termo dependente.

Na Figura 8, a árvore de dependência considera as categorias morfossintáticas de cada palavra. Diferentemente do trabalho de (CARROLL; CHARNIAK, 1992), as setas apresentam relação inversa.

A representação mais utilizada emprega o formalismo do *Universal Stanford dependencies* (MARNEFFE *et al.*, 2014). Essa representação é atualmente utilizada na UD.

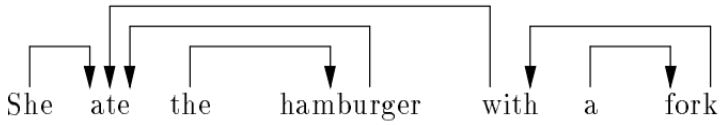


Figura 7 – Primeira representação de dependência aplicada em computação

Fonte: (CARROLL; CHARNIAK, 1992, Pag. 3)

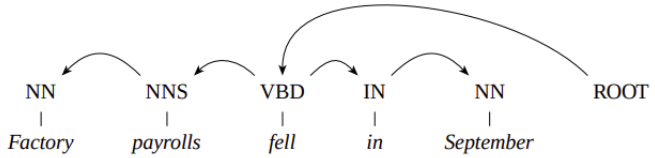


Figura 8 – Representação utilizando categorias morfossintáticas

Fonte: (KLEIN; MANNING, 2004, Pag. 2)

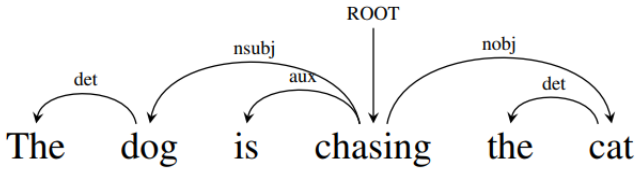


Figura 9 – Representação *Universal Stanford dependencies* (MARNEFFE *et al.*, 2014)

Fonte: (ZHU; BISK; NEUBIG, 2020, Pag. 649)

2.2 Indução gramatical

A IG utiliza o termo "gramática" em sua tarefa porque visa induzir as estruturas que compõem uma determinada gramática. No caso da gramática de constituintes, a IG busca induzir as regras responsáveis pela geração da árvore sintática, conforme descrito pela GLC. Já para a gramática de dependência, procura-se induzir as regras que governam as relações de dependência, como delineado no Capítulo 1.1 com base no trabalho de Hays (1964), que formalizou a gramática de dependência. Conforme apresentado na Seção 1.1, a tarefa de IG pode ser realizada de forma supervisionada, semi-supervisionada e não supervisionada.

Em aprendizado de máquina, a tarefa supervisionada aprende como associar uma entrada x (dado anotado) com uma saída y (resultado esperado). Os trabalhos de IGS utilizam dados com anotações sintáticas e morfossintáticas. Embora a IGS tenha alcançado resultados semelhantes a alguns trabalhos de IGNS (BOD, 2006a) e até tenha demonstrado "indícios" de que em breve poderia ser ultrapassada por IGNS (BOD, 2007), os estudos mais recentes de IGS para gramática de dependência mostram uma acurácia superior a 95% (LIN *et al.*, 2022), enquanto os melhores trabalhos de IGNS para gramática de dependência não conseguem atingir 70% de acurácia

(YANG *et al.*, 2020).

A tarefa semi-supervisionada, assim como a supervisionada, aprende como associar uma entrada x com uma saída y. No entanto, a entrada x é parcialmente anotada. Há dois tipos principais de treinamento semi-supervisionado. O primeiro, chamado de "*self-training*" (ROTMAN; REICHART, 2019), treina o modelo com uma pequena porção de dados anotados. Em seguida, esse modelo treinado é usado para treinar dados não anotados. O segundo tipo, denominado "*co-training*" (MAVELI; COHEN, 2022), consiste em dividir as características dos dados a serem treinados em dois classificadores e treiná-los com dados anotados e, posteriormente, com dados não anotados. Devido à proximidade entre tarefas semi-supervisionadas e não supervisionadas, alguns estudos semi-supervisionados são divulgados como se fossem não supervisionados. Esses estudos normalmente empregam termos como "semi-supervisão fraca" para descrever a influência supervisionada na tarefa de IG (MAVELI; COHEN, 2022).

Por fim, a tarefa não supervisionada, objeto deste trabalho, não utiliza qualquer tipo de anotação sintática, apesar de ser comum o uso de anotações morfossintáticas. Essa tarefa tem como objetivo tentar extrair informações dos dados. Uma vez que o trabalho de IGNS tenta induzir a gramática a partir da sintaxe, e não morfossintaxe, usualmente categorias morfossintáticas são utilizadas a fim de reduzir a esparsidade dos dados e complexidade computacional (KLEIN; MANNING, 2002) (III; JOHNSON; MCCLOSKY, 2009) (PARIKH; COHEN; XING, 2014). Alguns trabalhos de IGNS são mais rigorosos em não utilizar nenhum tipo de anotação no treinamento ou informação extra. Neste trabalho descrevemos este tipo de IG como "IGNS forte".

2.3 Métodos de indução gramatical

Há uma grande diversidade de métodos utilizados em IGNS. Descrever estes métodos nesta seção é impraticável, mesmo apenas os métodos utilizados nos 49 estudos selecionados descritos no Capítulo 3. Nesta seção, optamos por abordar os métodos mais comuns de maneira mais generalizada. Os métodos específicos utilizados em estudos individuais são descritos no Capítulo 3.

Os métodos aplicados em IGNS,tanto com gramática de constituintes quanto com gramática de dependência, dividem-se em duas categorias: gerativo e discriminativo. Todos os modelos discriminativos selecionados no mapeamento sistemático descrito no Capítulo 3, utilizam também o modelo gerativo ((CAI; JIANG; TU, 2017), (HAN; JIANG; TU, 2019) e (ANDREW, 2019)). Em aprendizado de máquina, os modelos gerativos modelam a probabilidade conjunta $P(x,y)$ entre a entrada x e a classe y que se pretende classificar. O modelo é dito gerativo porque gera exemplos a partir da distribuição de probabilidade de x. Por exemplo, o chatGPT, em termos gerais, é um modelo gerativo porque gera exemplos a partir de uma distribuição de probabilidade definida a partir de bilhões de palavras. Em IGNS, o modelo gerativo busca gerar exemplos de

árvores sintáticas que representam determinada sentença.

Os discriminativos modelam a probabilidade condicional $P(y|x)$, onde, dada a entrada x, estima-se qual a probabilidade de ocorrer a classe y. Em IGNS, a entrada x é a sentença e, a partir desta variável, busca-se encontrar a probabilidade de a árvore sintática y representar a sentença x. Neste tipo de estratégia, busca-se maximizar a probabilidade. Na Equação 2.1, a árvore sintática mais provável y^* para a sentença x, é dada pela maior probabilidade de todas as possíveis árvores sintáticas que podem representar a sentença x, representada por $\Psi(x)$.

$$y^* = \arg \max_{y \in \Psi(x)} P(y|x) \tag{2.1}$$

A maioria dos trabalhos de IGNS utiliza modelos gerativos. No mapeamento sistemático apresentado no Capítulo 3, constatamos que todos trabalhos selecionados utilizam modelos gerativos em alguma etapa do treinamento. Sendo apenas 3 estudos que utilizam modelos discriminativos. Uma das principais vantagens de utilizar um modelo gerativo em vez de um discriminativo é que o modelo gerativo é mais fácil de otimizar (MURPHY, 2022).

Há alguns trabalhos que não é possível descrever como discriminativos ou gerativos. Um exemplo de trabalho foi produzido por Bod (2006a), que seleciona aleatoriamente um conjunto de árvores possíveis para representar a sentença analisada a partir de uma estimativa das sub-arvores que compõem a árvore principal. Por sua vez, Seginer (2007) constrói a árvore sintática de forma incremental antes de analisar toda a sentença utilizando heurísticas. Por fim, Bod (2006b) propôs uma generalização do trabalho produzido por Bod (2006a).

2.3.1 Abordagem Gerativa

A abordagem gerativa pode ser categorizada em três grupos: estrutural, não paramétrica e paramétrica (TU *et al.*, 2021). Na abordagem gerativa estrutural, por vezes referida como "busca estrutural" ou "aprendizado estrutural", não existe uma gramática fixa para o modelo. Em vez disso, ela é aprendida juntamente com os parâmetros do modelo, utilizando-se heurísticas.

Entre as principais técnicas estão *Hidden Markov Models* (HMM) (técnica utilizada para predizer sequências) (STOLCKE; OMOHUNDRO, 1994) e *Latent Tree Graphical Model* (LTGM) (árvore que contém variáveis observáveis e não observáveis) (PARIKH; COHEN; XING, 2014), para citar algumas. Os primeiros trabalhos em IG geralmente utilizavam a abordagem estrutural. O trabalho de (KLEIN; MANNING, 2002) contribuiu para a mudança de paradigma do modelo estrutural para o paramétrico em IGNS.

Modelos paramétricos têm um número fixo de parâmetros, como, por exemplo, μ (média) e σ^2 (variância). No entanto, não é correto afirmar que os modelos não paramétricos não possuam parâmetros, uma vez que não é possível determinar o número de parâmetros nesses modelos. Isso ocorre porque o número de parâmetros aumenta conforme a quantidade de dados cresce,

podendo, teoricamente, ser infinito (GERSHMAN; BLEI, 2012). Essa abordagem tem como vantagem permitir ajustar o modelo a sua complexidade e quantidade de dados. Alguns trabalhos sugerem que Seginer (2007) e Bod (2006a) são não parametrizados por não utilizarem parâmetros. Outros sugerem que são gerativos estruturais (SANKARAN, 2010). No entanto, neste estudo, adotamos uma abordagem mais rigorosa ao seguir a definição matemática de modelo não paramétrico. Portanto, não os categorizamos como gerativos ou não paramétricos, deixando assim em aberto a classificação de ambos os trabalhos. Uma técnica bastante utilizada em tarefas de IGNS com abordagem não paramétrica é o uso de inferência Bayesiana (COHN; BLUNSOM; GOLDWATER, 2010).

Por fim, a abordagem paramétrica é a mais amplamente utilizada em IGNS, respondendo por mais de 92% dos trabalhos gerativos, de acordo com nossa pesquisa apresentada no Capítulo 3. A abordagem parametrizada inicia com um modelo de gramática fixo. Os parâmetros do modelo são então otimizados a fim de encontrar a melhor árvore sintática para uma determinada sentença. As principais técnicas utilizadas na otimização dos parâmetros são a inferência Bayesiana, os algoritmos IO (que pode ser visto como uma instância do EM) (BAKER, 1979) e EM (DEMPSTER; LAIRD; RUBIN, 1977), e redes neurais (incluindo modelos de linguagem).

2.3.1.1 *Inferência Bayesiana*

A inferência Bayesiana é usada quando temos variáveis não observáveis. Pois, o uso da regra de Bayes (ou teorema de Bayes), permite inferir uma determinada distribuição desconhecida a partir de outras distribuições conhecidas. Considerando que IGNS trata as árvores sintáticas como variáveis não observáveis, a inferência Bayesiana é uma boa estratégia para inferir a distribuição dessa variável, principalmente quando pode ser tratada como a posteriori (informação desejada, (MURPHY, 2022)) ou a priori (informação prévia antes da obtenção dos dados do modelo, (MURPHY, 2022)).

A inferência dessas distribuições pode ser calculada através de aproximação. Há duas estratégias principais: amostragem ou *Variational Inference* (VI) (BLEI; KUCUKELBIR; MCAULIFFE, 2017). Ambas as estratégias fazem parte da "caixa de ferramentas" de inferência Bayesiana e são constituídas de técnicas computacionais para tal tarefa. A técnica de amostragem é principalmente conhecida a partir do uso de *Chain Monte Carlo Sampling* (MCMCS) com variantes como amostragem de Gibbs e *Metropolis-Hastings algorithm*. Tanto VI quanto MCMCS são utilizados para estimar a *posteriori* (MAREČEK; ŽABOKRTSKỲ, 2011). Em alguns casos, utiliza-se *Dirichelet* e *Logistic normal* em VI para também estimar a *priori*. MCMCS apresenta a vantagem sobre VI por ser um método garantido de aproximação, apesar de ser computacionalmente custoso (BLEI; KUCUKELBIR; MCAULIFFE, 2017).

Conforme já descrevemos, a gramática gerativa é utilizada para calcular a probabilidade conjunta $P(y,x)$. O teorema de Bayes é utilizado para calcular a probabilidade conjunta a partir do cálculo da *posteriori* $P(y|x)$ ou da *priori* $P(y)$, a depender do tipo de distribuição que é

utilizada (COHEN; GIMPEL; SMITH, 2008). A *Marginal Likelihood* $P(x)$, que é responsável por normalizar a distribuição de probabilidade, não é geralmente utilizada para computar a probabilidade conjunta, pois, em certos casos, é bastante complicada de ser calculada (ROGERS; GIROLAMI, 2016). O teorema de Bayes é apresentado na Equação 2.2.

$$P(x,y) = P(y|x)P(x) \rightarrow P(y|x) = \frac{P(x,y)}{P(x)} = \frac{P(x|y)P(y)}{P(x)} \tag{2.2}$$

Para estimar a *priori*, geralmente são utilizadas as distribuições *Dirichlet* e *Logistic Normal* (COHEN; GIMPEL; SMITH, 2008). A distribuição *Dirichlet* é uma generalização da distribuição *Beta* (MURPHY, 2022). Essa distribuição pode ser vista como uma distribuição de distribuições. A *Dirichlet* é uma distribuição de densidade, o que significa que a soma de todos os valores deve resultar em 1. A função de distribuição de densidade é dada pela Equação 2.3.

$$f(x_1,x_2,...,x_k;\theta_1,...,\theta k) = \frac{\Gamma(\sum_{i=1}^{k}\theta_i)}{\prod_{i=1}^{k}\Gamma(\theta_i)}\prod_{i=1}^{k}x_i^{\theta_i-1} \tag{2.3}$$

Onde $0 \leqslant x \leqslant 1$ e $\sum_{i=1}^{k} x_k = 1$, θ são as parâmetros e $\Gamma(\theta_i)$ é a função gama[1]. Quando o número de parâmetros é indeterminado, temos então um *Dirichlet Process* (processo de *Dirichlet)* (MURPHY, 2022), utilizado em IGNS não paramétrica.

A *Logistic-normal* também é uma distribuição de distribuições. No entanto, ela é pouco explorada em tarefas de indução gramatical. Enquanto é possível encontrar alguns trabalhos de IGNS que utilizam a distribuição *Dirichlet* sem o uso de VI (JIN *et al.*, 2018) (JIN *et al.*, 2019), não encontramos nenhum trabalho no mapeamento sistemático que utilize *Logistic Normal* de forma independente. A distribuição *Logistic-normal* também pode ser utilizada para modelos não paramétricos por meio de sua discretização, denominada de *Discrete Infinite Logistic Normal* (PAISLEY; WANG; BLEI, 2011).

VI é um método de aprendizado de máquina utilizado para aproximar uma distribuição (BLEI; KUCUKELBIR; MCAULIFFE, 2017). Em comparação com MCMC, VI é consideravelmente mais rápido e mais facilmente escalável para grandes volumes de dados, o que explica sua preferência dentro da abordagem Bayesiana em tarefas de IGNS[2]. No entanto, é importante notar que VI é um método complexo de aplicar e compreender (SPITKOVSKY; ALSHAWI; JURAFSKY, 2010) (BLEI; KUCUKELBIR; MCAULIFFE, 2017). A utilização de otimização é a principal ideia de VI. A otimização é realizada a partir da aproximação de densidades de probabilidade Ξ. A ideia é encontrar um membro ϖ de uma família de probabilidades de forma que minimize a distância *Kullback-Leibler* (KL) (distância que mede o quão diferente são duas distribuições (CSISZÁR, 1975)) da posteriori $P(y|x)$.

[1] $\Gamma(\theta_i) = (\theta_i - 1)!$

[2] Para mais detalhes, consulte o Capítulo 3

$$\arg \max_{\varpi(y) \in \Xi} KL(\varpi(y) : P(y|x)) \tag{2.4}$$

O MCMC é uma alternativa ao uso do VI. A ideia do MCMC é gerar amostras aleatórias de uma variável aleatória de uma distribuição de densidade. O princípio deste método é gerar amostras aleatórias até que a distribuição de densidade seja completamente explorada (método de Monte Carlo) (BLEI; KUCUKELBIR; MCAULIFFE, 2017). Os métodos mais populares utilizados são *Metropolis-Hastings algorithm* e amostragem de Gibbs (BLEI; KUCUKELBIR; MCAULIFFE, 2017). O último é mais comum em tarefas de IGNS que, em vez de gerar observações individuais, gera sequências de observações, o que é útil para tarefas de IGNS.

2.3.1.2 Expectation Maximization

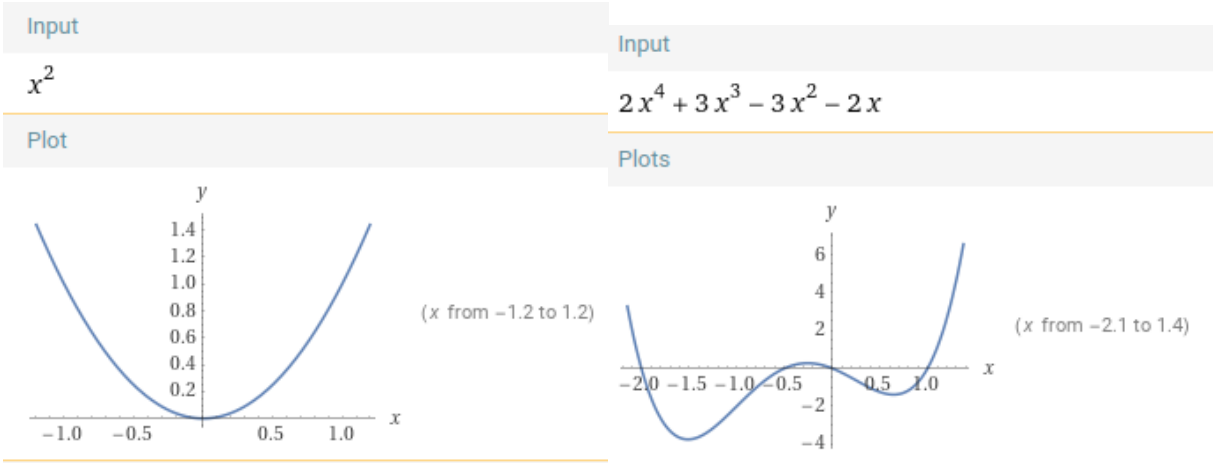
O algoritmo EM, e suas variações, é o mais usado em tarefas de IGNS, correspondendo a mais de 46% dos estudos em IGNS, segundo o mapeamento sistemático apresentado no Capítulo 3. O EM é um algoritmo iterativo que tem utilidade em problemas em que há falta de dados ou que os dados não sejam observáveis (MCLACHLAN; KRISHNAN, 2007). A ideia deste algoritmo é maximizar os parâmetros de modo que estes sejam representativos da distribuição de probabilidade (*Maximum Likelihood* (ML)) e, dessa forma, permitir estimar os dados não observáveis.

O algoritmo é dividido em duas partes: *Expectation* (etapa E, descrita na Equação 2.5) e *Maximization* (etapa M, descrita na Equação 2.6) . Na etapa **E**, o algoritmo estima os dados não observáveis. Em IGNS, os dados observáveis são as sentenças x e os não observáveis são as árvores sintáticas y. A escolha dos parâmetros depende do projeto do modelo. Geralmente os parâmetros μ e σ^2 são utilizados. Na etapa **E**, é calculada a *posteriori* (Equação 2.5), onde Θ refere-se aos parâmetros do modelo. A etapa **M** tem como objetivo maximizar os parâmetros para que seja possível estimar os dados não observáveis na etapa **E**. Em IGNS, a etapa **M** geralmente é utilizada para otimizar a *Marginal Likelihood*. Em alguns trabalhos, *log likelihood* é utilizada devido a computação logaritma do resultado.

$$E : Step \rightarrow P(y|x, \Theta) \tag{2.5}$$

$$M : Step \rightarrow log \sum_{Y=y} P(X, y) \tag{2.6}$$

Uma vez que os parâmetros do algoritmo precisam ser inicializados, são usados tanto valores fixos quanto aleatórios. EM é bastante utilizado para otimizar funções não convexas. Esse tipo de função pode apresentar mais de um máximo local (pontos de inflexão da função), conforme apresentado na Figura 10(b), com dois máximos locais. A Figura 10(a) apresenta uma função convexa, isto é, existe apenas um máximo local.



(a) Função convexa (b) Função não convexa

Figura 10 – Exemplo de funções

Fonte: Figuras geradas através do site https://www.wolframalpha.com/

Otimizar funções convexas é relativamente simples, pois requer apenas a derivada igualada a zero. No entanto, otimizar funções não convexas é bastante complicado devido aos máximos locais (BARAZANDEH; RAZAVIYAYN, 2018). Essa é uma das dificuldades encontradas com o EM, o que motivou o desenvolvimento de variantes para mitigar essa desvantagem. Entre as variantes discutidas neste estudo estão o *Lateem-EM* (SPITKOVSKY; ALSHAWI; JURAFSKY, 2011a), a *regularização posterior* (GILLENWATER *et al.*, 2011) e o *Viterbi-EM* (SPITKOVSKY *et al.*, 2010). Esses algoritmos são abordados com mais detalhes no Capítulo 3.

2.3.1.3 Redes neurais

Apesar de redes neurais existirem há décadas, não havia um modelo neural que pudesse induzir gramática, de forma não supervisionada, melhor que os modelos que utilizam inferência Bayesiana ou EM (SHEN *et al.*, 2018). As redes mais comumente utilizadas na construção dos modelos de IGNS são convolucionais, recorrentes e perceptrons de multicamadas. As redes neurais são geralmente utilizadas para estimar a probabilidade conjunta (JIANG; HAN; TU, 2016; ZHU; BISK; NEUBIG, 2020). Um modelo de rede neural bastante utilizado em atividades de IGNS é apresentado na Figura 11.

O modelo recebe como entrada a ***Valency***[3], a cabeça da relação, ***Head Word***, e a categoria morfossintática da cabeça da relação, ***Head POS tag***. A saída da rede gera os possíveis ramos ***CHILD Outputs*** e a probabilidade de um termo para de ser gerado,***DECISION Outputs***, conforme o modelo DMV apresentado no Capítulo 1.

Os modelos de linguagem já eram usados antes mesmo do surgimento as redes neurais com *backpropagation* (SUEN, 1979; RUMELHART *et al.*, 1985). No entanto, o uso de modelos

[3] O número e tipo do argumento controlado pelo predicado, entre os mais comuns estão os verbos transitivos e intransitivos (ALLERTON, 1982)

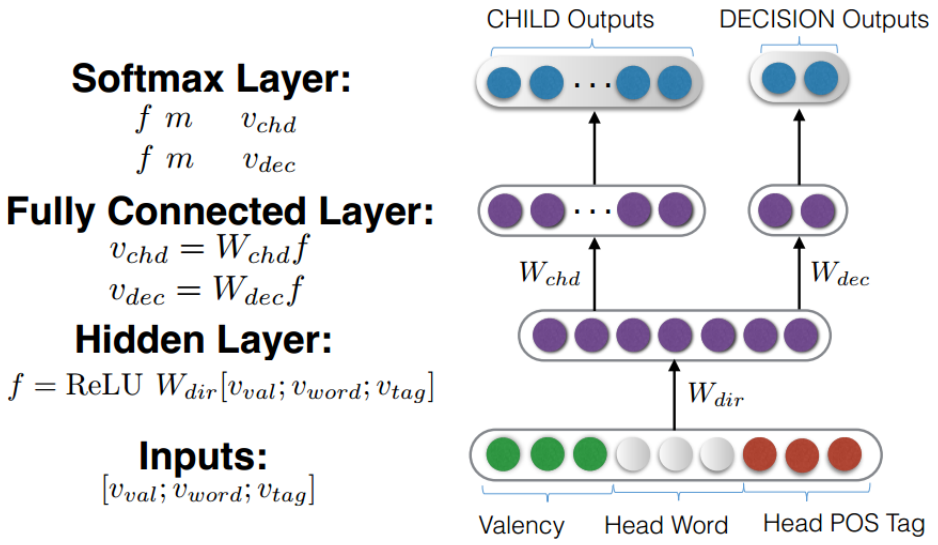


Figura 11 – Rede Neural aplicado ao modelo DMV (KLEIN; MANNING, 2004)

Fonte: (HAN; JIANG; TU, 2017, Pag. 2)

de linguagem era pouco explorado na tarefa de IGNS. No mapeamento sistemático, que apresentamos no Capítulo 3, encontramos poucos estudos que utilizavam modelos de linguagem aplicados a IGNS sem uso de redes neurais, sendo Chen (1995) o mais relevante deles. Com o surgimento dos *transformers* (VASWANI *et al.*, 2017), os modelos de linguagem passaram a ser utilizados com mais frequência em IGNS. Uma das principais estratégias é utilizar modelos pré-treinados para extrair estruturas sintáticas. Os estudos que buscavam (e ainda buscam) atestar se os modelos de linguagem compreendem sintaxe, exerceu bastante influência para o uso em IGNS (KIM *et al.*, 2020).

2.3.2 Abordagem Discriminativa

A abordagem discriminativa modela a probabilidade condicional $P(y|x)$, onde dada uma sentença x, busca-se computar a probabilidade de a árvore sintática y representar a sentença x. Os primeiros trabalhos discriminativos utilizavam técnicas de agrupamento (CLARK, 2001). No entanto, nos últimos 20 anos desde a publicação desse trabalho, pouquíssimos trabalhos que aplicam esta técnica foram utilizados para IGNS. Os trabalhos mais recentes utilizam *autoencoder*.

Autoencoder é um tipo de rede neural que tenta reproduzir na saída os mesmos valores inseridos na entrada (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa rede é dividida em duas partes: *encoder* (codificador) e *decoder* (decodificador). O cálculo das distribuições de probabilidade depende do método adotado. Geralmente, o *encoder* computa a probabilidade condicional $P(y|x)$ e o *decoder* a probabilidade condicional $P(x|y)$. Alguns trabalhos utilizam uma abordagem combinada ao implementar um método discriminativo no *encoder* e, generativo

no *decoder* (LI *et al.*, 2019). A Figura 12 apresenta um modelo de *autoeconder* aplicado à gramática de dependência.

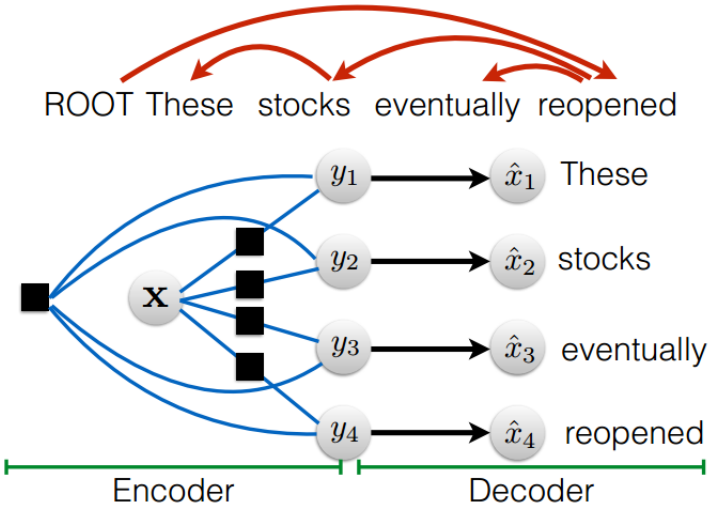


Figura 12 – Autoencoder para indução gramatical

Fonte: (CAI; JIANG; TU, 2017, Pag. 2)

A ideia central no uso do *autoencoder* é que, no intermediário da rede, é computada a árvore sintática que representa a sentença gerada na saída (CAI; JIANG; TU, 2017). Outra forma de aplicar treinamento discriminativo é por meio da técnica de agrupamento. No mapeamento sistemático apresentado no Capítulo 3, todos os estudos selecionados aplicam *autoencoder*. Apesar de Clark (2001) utilizar agrupamento no treinamento discriminativo e ser um importante trabalho para a área de IG, esse trabalho não foi recuperado no mapeamento sistemático.

2.4 Córpus

Os corpora que contêm anotações de árvores sintáticas provenientes de algum tipo de gramática, como gramáticas de constituintes ou de dependência, são comumente conhecidos como *Treebank* (banco de árvores), conforme exemplificado nas Figuras 13 (árvore de constituintes) e 14 (árvore de dependência). Nas tarefas de IGNS para gramática de dependência, geralmente não se utilizam *Treebank*, a menos que seja necessário comparar os resultados com outros estudos que tenham utilizado *Treebank*.

O formato CONLL (BUCHHOLZ; MARSI, 2006) vem ganhando bastante relevância nos trabalhos de indução gramatical de múltiplas línguas, principalmente para gramática de dependência. Esse é o formato adotado pelo *framework* UD, que será apresentado na subseção seguinte. O CONLL é apresentado na Figura 15.

O CONLL representado na Figura 15, utiliza 10 colunas.A primeira coluna descreve o ID do *token* (menor unidade do texto não separável, seja palavra, pontuação ou símbolo). Podemos

```
( (S
   (NP Battle-tested industrial managers
       here)
    always
   (VP buck
       up
       (NP nervous newcomers)
       (PP with
           (NP the tale
               (PP of
                   (NP (NP the
                           (ADJP first
                                 (PP of
                                     (NP their countrymen)))
                           (S (NP *)
                              to
                              (VP visit
                                  (NP Mexico))))
                       ,
                       (NP (NP a boatload
                               (PP of
                                   (NP (NP warriors)
                                       (VP-1 blown
                                           ashore
                                           (ADVP (NP 375 years)
                                                 ago)))))
                           (VP-1 *pseudo-attach*)))))))
  .)
```

Figura 13 – Extrato de *Treebank* de árvore de constituintes para a sentença "*Battle-tested industrial managers here always buck up nervous newcomers with the tale of the first of their countrymen to visit Mexico, boatload of samurai warriors blown ashore 375 years ago.*"

Fonte: (MARCUS; SANTORINI; MARCINKIEWICZ, 1993, pag. 13)

```
STA:fcl
=SUBJ:np
==H:n('enchente' <np-idf> F P)  Enchentes
=P:vp
==MV:v-fin('dar' <se-passive> <hyfen> PR 3P IND)  dão-
=ACC-PASS:np
==H:pron-pers('se' <refl> F 3P ACC)  se
=ADVL:pp
==H:prp('por' <sam->)  por
==P<:np
===>N:art('o' <-sam> <artd> M S)  o
===H:n('país' <np-def> M S)  país
===N<:adjp
====H:adj('inteiro' M S)  inteiro
=.
```

Figura 14 – Extrato de *Treebank* de árvore de dependência para a sentença "Enchentes dão-se pelo país inteiro"

Fonte: (AFONSO *et al.*, 2002, Bosque CETENFolha n=878)

observar que as linhas 2 e 4 são aglutinadas com outro token. A linha 2 aglutina os *tokens* "dão se" e a linha 4 aglutina os *tokens* "por o". Considerando a replicação da informação, a aglutinação

```
1     Enchentes  enchente  NOUN   _  Gender=Fem|Number=Plur                          2  nsubj  _  _
2-3   dão-se     _         _      _  _                                               _  _      _  _
2     dão        dar       VERB   _  Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin  0  root   _  _
3     se         se        PRON   _  Case=Acc|Gender=Fem|Number=Plur|Person=3|PronType=Prs  2  expl   _  _
4-5   pelo       _         _      _  _                                               _  _      _  _
4     por        por       ADP    _  _                                               6  case   _  _
5     o          o         DET    _  Definite=Def|Gender=Masc|Number=Sing|PronType=Art  6  det    _  _
6     país       país      NOUN   _  Gender=Masc|Number=Sing                         2  obl    _  _
7     inteiro    inteiro   ADJ    _  Gender=Masc|Number=Sing                         6  amod   _  SpaceAfter=No
8     .          .         PUNCT  _  _                                               2  punct  _  _
```

Figura 15 – Extrato de córpus de árvore de dependência para a sentença “Enchentes dão-se pelo país inteiro”

Fonte: (AFONSO *et al.*, 2002, Bosque CETENFolha n=878)

resultante pode ser empregada ou não na execução da tarefa de IGNS. No presente estudo, optamos por remover todos os tokens aglutinados. As segunda e terceira colunas descrevem respectivamente o *token* do texto anotado e o lema da palavra. A coluna 4 descreve a categoria morfossintática. As colunas 5 e 10 são para casos específicos de determinada língua. A coluna 6 descreve as características da categoria morfossintática. Por fim, as colunas 7, 8 e 9 descrevem, respectivamente, a cabeça da relação da qual o *token* da linha depende, o tipo de relação existente e uma outra forma de representação de dependente.

2.4.1 Córpus para o Português

O Bosque foi construído a partir do projeto *Floresta Sinta(c)tica* (AFONSO *et al.*, 2002), o primeiro *treebank* construído para a língua portuguesa para gramática de dependência e gramática de constituintes. Isso contribuiu para que o Português fosse a língua mais utilizada para gramática de dependência depois do chinês, inglês e alemão, segundo o mapeamento sistemático descrito no Capítulo 3. Tanto o Bosque para gramática de constituintes quanto para gramática de dependência utilizam as mesmas sentenças. O córpus para gramática de constituintes pode ser baixando no site do projeto Floresta Sintáctica. Já a versão em gramática de dependência (RADEMAKER *et al.*, 2017) pode ser baixada no site do projeto UD. Há também outros 3 córpus para a gramática de dependência reconhecidos: Petrogold (SOUZA *et al.*, 2021), CINTIL, (BRANCO *et al.*, 2022), *Parallel UD* (PUD) (ZEMAN *et al.*, 2018) e Porttinari(PARDO *et al.*, 2021).

O Bosque contém 210k *tokens* e 9k sentenças, sendo 55% das sentenças originárias do córpus CETEMPúblico (ROCHA; SANTOS, 2000), construído com textos jornalísticos do português europeu, e 45% do CETENFolha (LINGUATECA, 2023), construído com textos jornalísticos do português brasileiro. O córpus PetroGold é formado por 19 teses e dissertações na área de óleo e gás escritos em português do Brasil, constituído por 232k *tokens* e 9k sentenças. O CINTIL contém textos exclusivamente em português europeu extraídos tanto da língua falada quanto da língua escrita de jornais, livros, revistas e conteúdos informais. O PUD é um corpús

paralelo que consiste de 1000 sentenças e 21k *tokens* traduzidas em 18 línguas. Dos córpus apresentados, apenas o Porttinari utiliza textos informais produzidos na Web. O Porttinari conta com textos jornalísticos, diferente dos utilizados pelo Bosque e CINTIL, e textos da web, que compreendem *review* de produtos e mensagens da plataforma *Twitter* sobre o mercado financeiro. O detalhamento dos córpus é apresentado na Tabela 1.

Tabela 1 – Descrição dos principais córpus anotados em língua portuguesa

Córpus	**Origem**	**T**	**V**	**S**	**μ T p/S**	**S $\leqslant$ 3T**	**S > 40T**
Bosque	BR/PT	210958	26437	9357	26,15	3,59%	18,59%
CINTIL	PT	441991	34498	38400	12,39	1,09%	1,31%
Petrogold	BR	232333	15528	8945	30,05	4,49%	24,70%
Porttinari	BR	176775	19188	8419	20,99	0%	4,03%
PUD	BR/PT	21917	5964	1000	24,88	0,10%	7,51%

O tipo de português que compõe os corpus é descrito na segunda coluna. Para sentenças produzidas em português europeu – PT e para sentenças produzidas em português brasileiro – BR. As letras T, V e S representam respectivamente as palavras *tokens*, vocabulário e sentença.

2.4.2 Córpus anotados para línguas estrangeiras

O *Prague Dependency Treebank* foi o primeiro *Treebank* para gramática de dependência (HAJIČ, 1998). Ele foi construído para a língua Tcheca e contém incríveis 3030 tags morfossintáticas e sintáticas. Dos 17 estudos selecionados na revisão sistemática que utilizam a língua Tcheca, 16 são para gramática de dependência. No entanto, o *Prague Dependency Treebank* não se tornou um *Benchmark*.

O PTB (MARCUS; SANTORINI; MARCINKIEWICZ, 1993) é um córpus anotado com árvores sintáticas de gramática de constituintes. Apesar disso, ele se tornou um *Benchmark* não apenas para gramática de constituintes, mas também para gramática de dependência. Mais especificamente, as seções do PTB *Wall Street Journal* (WSJ) é utilizada para IG, pois foram anotadas por especialistas. Os primeiros trabalhos relevantes de IGNS para gramática de dependência utilizaram o WSJ (KLEIN; MANNING, 2004; SMITH; EISNER, 2005). Para poder comparar os resultados do modelo com uma referência, eles utilizaram o algoritmo proposto por Collins (2003), que converte gramática de constituintes para gramática de dependência. O WSJ completo é composto por pouco mais de 1 milhão de *tokens*. A relação do número de sentenças com o número de *tokens* do córpus, assim como a quantidade de sentenças e categorias morfossintáticas, foi apresentada por Spitkovsky, Alshawi e Jurafsky (2010).

Dois outros corpus bastante utilizados em tarefas de IGNS são o CTB (XUE *et al.*, 2005), anotado para a língua chinesa, e o NEGRA (SKUT *et al.*, 1998), anotado para a língua alemã. Há também os córpus multilíngues. O primeiro esforço para a criação de córpus multilíngue foi apresentando no *CoNLL-X shared task on Multilingual Dependency Parsing*, que teve

como tarefa, converter 7 *treebanks* de gramática de constituintes em gramática de dependência (BUCHHOLZ; MARSI, 2006).

2.5 *Universal Dependencies*

No Capítulo 1, apresentamos que a proposta da UD é ousada em tentar propor uma gramática de dependência em comum, universal. No entanto, a proposta não está relacionada com a Gramática Universal proposta por Chomsky. Segundo Nivre (2015), um dos idealizadores da UD, não é objetivo da UD corroborar com hipóteses e explanações sobre as estruturas das línguas . Em vez disso, a UD busca explorar as semelhanças entre as línguas. Essas características são divididas em morfossintáticas e sintáticas. As características morfossintáticas são separadas em 17 categorias. Uma dessas categorias é utilizada para representar diferenças morfossintática entre as línguas, assim como palavras não catalogadas ou pertencentes à língua. Além das 17 categorias, foram definidos também 215 atributos para as categorias, como (gênero, número, grau, modo, tempo, para citar alguns). A lista completa pode ser verificada no site https://universaldependencies.org/u/feat/index.html). As características sintáticas são descritas em 37 categorias. Essas categorias comportam ainda 75 subgrupos que podem ser verificados no site https://universaldependencies.org/u/dep/index.html.

2.5.1 *Morfossintaxe*

Antes da existência de um *framework* que visava representar o máximo de línguas possível utilizando o menor número de categorias morfossintáticas, muitos *frameworks* empregavam dezenas, e por vezes centenas, de categorias morfossintáticas. O *framework* PTB, o mais utilizado na tarefa de IG, utiliza 48 categorias morfossintáticas.

Uma das estratégias utilizadas pela UD para tornar mais consistente a gramática entre as línguas é segmentar palavras aglutinadas. Em línguas latinas, a aglutinação é bem comum. Na língua portuguesa, por exemplo, há a aglutinação de preposição com artigo ($do \Rightarrow de + o$). No *framework* UD, essas palavras são tratadas de forma separada, mas também podem, no mesmo córpus, conter as duas versões.

A UD foi concebida com 17 categorias morfossintáticas que foram baseadas no *Google Universal Part-of-Speech Tagset* (PETROV; DAS; MCDONALD, 2012). As categorias estão apresentadas na Tabela 2. Desde sua concepção oficial, com a proposta do primeiro *treebank* para gramática de dependência por (MCDONALD *et al.*, 2013), que compartilha relações sintáticas entre diversas línguas, até a versão mais recente (NIVRE *et al.*, 2020), houve apenas uma pequena alteração: da categoria CONJ para CCONJ. Essas sutis variações do *treebank* entre diferentes versões contribuem para a consistência e padronização dele. Por exemplo, a categoria **X** é utilizada quando não é possível informar gramaticalmente a que categoria a palavra pertence: palavras de outras línguas, letras gregas utilizadas na matemática, palavras

Tabela 2 – Categorias morfossintáticas da UD

Categoria	**Descrição**
ADJ	Adjetivo
ADP	Adposição
ADV	Advérbio
AUX	Verbo auxiliar
CCONJ	Conjunção Coordenativa
DET	Determinante
INTJ	Interjeição
NOUN	Substantivo
NUM	Numeral
PART	Particípio
PRON	Pronome
PROPN	Pronome próprio
PUNCT	Pontuação
SCONJ	Conjunção subordinativa
SYM	Símbolo
VERB	Verbo
X	Outro

com erros de ortografia, erros no processo de tokenização (i.e., separação das unidades do texto) e demais palavras da língua que o anotador não conseguiu identificar. Para essas palavras, utiliza-se a categoria **X**. Nos córpus da língua portuguesa utilizados neste trabalho (Bosque, Porttinari e Petrogold), 87% das palavras anotadas com a categoria morfossintática **X** são palavras estrangeiras.

2.5.2 Sintaxe

O *framework* UD contém 37 relações sintáticas. Essas relações sintáticas foram baseadas no *Universal Stanford Dependencies* (MARNEFFE *et al.*, 2014), que originalmente continha 40 relações. As 37 relações são divididas em dois grupos: dependências funcionais e dependências não funcionais. O grupo de relações sintáticas com dependências funcionais é apresentada na Tabela 3.

As categorias estruturais separam as relações pelo tipo de estrutura sintática com a qual se relacionam. Há quatro categorias estruturais: *nominals*, *clauses*, *modifier words* e *function words*. *Nominals* são as categorias que apresentam relações com nomes (substantivo, pronome e frase substantiva). Entre as relações sintáticas *nominals* mais frequentes estão *nsubj* (sujeito), *nmod* (modificadores de nomes) e *obl* (nome oblíquo). *Clauses* representam as orações da sentença, podendo ser *csubj* (oração subjetiva), *ccomp* (oração complementar com sujeito interno, sempre usada no infinitivo) e *xcomp* (oração complementar com sujeito externo) (MARNEFFE *et al.*, 2021). As categorias *modifier words* e *function words* descrevem as relações que modificam

Tabela 3 – Relações sintáticas de dependência funcionais

Categorias Funcionais	Categorias Estruturais			
	Nominals (Nominais)	*Clauses* (Orações)	*Modifier Words* (Modificadores)	*Function Words* (Palavras Funcionais)
Core Arguments (Argumentos principais)	*nsubj* *obj* *iobj*	*csubj* *ccomp* *xcomp*		
Non-Core dependents (Argumentos não principais)	*obl* *vocative* *expl* *dislocated*	*advcl*	*advmod** *discourse*	*aux* *cop* *mark*
Nominal Dependents (Dependentes nominais)	*nmod* *appos* *nummod*	*acl*	*amod*	*det* *clf* *case*

Tabela 4 – Relações sintáticas de dependências não funcionais

Coordination (coordenação)	*MWE* (expressão multipalavra)	*Loose* (informalidade)	Special (especial)	*Other* (outra)
conj *cc*	*fixed* *flat* *compound*	*list* *parataxis*	*orphan* *goeswith* *reparandum*	*punct* *root* *dep*

palavras e as relações que concedem funcionalidades às palavras (determinadores, por exemplo), respectivamente. Raramente um córpus anotado utiliza todas as relações.

O segundo grupo, apresentado na Tabela 4, contém cinco tipos de categorias. A categoria *coordination* representa as relações coordenativas. A categoria *MWE* representa as expressões multipalavras. Expressões multipalavras são expressões formadas por mais de uma palavra que pode ter sentido conotativo (“ovelha negra”: pessoa fora do padrão) ou denotativo (“ovelha negra”: ovino de cor negra) (SAG *et al.*, 2002). A relação *list* em *Loose* representa relações informais no texto escrito como lista de itens, e-mails e numerações. Já a relação *parataxis* representa relações um pouco mais formais, geralmente iniciada após parentese, vírgula ou dois pontos. A categoria *special* casos não muito comuns na língua, como por exemplo, o uso de elipse. Essa categoria representa menos de 0.004% de todas as relações. A última categoria representa pontuações, *root* (raiz da árvore de dependência) e quando não é possível definir com precisão o tipo de relação.

Raramente todas as categorias são utilizadas. O CINTIL, por exemplo, utiliza apenas 24 destas relações. Já o Petrogold utiliza 35 (apenas as relações *dep* e *clf* não são usadas, pois estas são destinadas para gramáticas específicas). O site https://universaldependencies.org/guidelines.html apresenta explação mais detalhada sobre as diferentes categorias e relações sintáticas.

2.6 Avaliação

A comparação de estruturas induzidas com as anotadas por humanos é o principal meio em que as tarefas de indução gramatical são avaliadas, conforme apresentado no capítulo 3. Outra alternativa é o uso de avaliação por humanos para sentenças novas (SOLAN *et al.*, 2003). As métricas utilizadas para avaliar tarefas de indução gramatical automática varia conforme o tipo de gramática.

Para as gramáticas de constituintes a mais comum é o uso de PARSEVAL (BLACK *et al.*, 1991) (SANKARAN, 2010). Dados CMPC (constituintes do modelo preditos corretamente), CM (constituintes do modelo), CTPC (constituintes do *treebank* preditos corretamente) e CT (constituintes do *treebank*), temos

$$precisao = \frac{CMPC}{CM}$$
$$recall = \frac{CTPC}{CT}$$

aplicados ao F1. A ferramenta que implementa o *PARSEVAL* é o *evalb* (SEKINE; COLLINS, 1997) disponível em https://nlp.cs.nyu.edu/evalb/. O problema desta métrica, é que ela pune *treebanks* com árvores não binárias (D'ULIZIA; FERRI; GRIFONI, 2011a). Portanto, é preciso binarizar todas as árvores usando a FNC para evitar penalização

Apesar de alguns estudos utilizarem F1 para avaliar gramática de dependência (SMITH; EISNER, 2006); (KLEIN; MANNING, 2004), as principais métricas utilizadas são a *Direct Dependency Accuracy* (DDA), *Undirected Dependency Accuracy* (UDA) e *Unlabeled Attachment Score* (UAS) . A métrica DDA avalia se a árvore produzida é exatamente que a árvore anotada, considerando a direção, caso não considere a direção então considera-se a UDA (KLEIN, 2005). A UAS é mais comum em IG supervisionada, pois para IG não supervisionada pode apresentar um forte bias (MAREČEK; ŽABOKRTSKỲ, 2011).

Alguns autores utilizam o *Directed Attachment Accuracy* (DAA) e *Undirected Attachment Accuracy* (UAA) como métricas para realizar o mesmo tipo de avaliação de DDA e UDA, respectivamente.

A gramática de constituintes tem como *baselines* os modelos CCM (KLEIN; MANNING, 2002) e o *Unsupervised Data-Oriented language Processing* (UDOP) (BOD, 2006a). Para gramática de dependência são usados o *baseline* ramificação direita (*right-branching*) e o DMV (KLEIN; MANNING, 2004). O *right-branching* é um modelo que utiliza sempre a primeira palavra da sentença como raiz e a palavra da direita sempre será dependente da palavra à esquerda (PATE; JOHNSON, 2016).

CAPÍTULO

3

UNSUPERVISED GRAMMAR INDUCTION IN NATURAL LANGUAGE PROCESSING: A SYSTEMATIC MAPPING STUDY

O trabalho intitulado *"Unsupervisded Grammar Induction in Natural Language processing: A systematic mapping survey"* está em processo de revisão para ser submetido a uma revista. Este trabalho foi motivado pela necessidade em buscar os principais trabalhos relevantes na área. Apesar de haver algumas revisões sistemáticas, e *reviews* sobre a área, essas referem-se à apenas alguns temas em específico e não apresentam uma busca profunda na área. A principal contribuição deste artigo neste trabalho de mestrado foi apresentar um aprofundado mapeamento sistemático sobre a área de indução gramatical de forma a compilar trabalhos de vários temas dentro da indução gramatical. Nesse trabalho, foi realizado não apenas uma sumarização da área, mas também uma extensa análise sobre os principais métodos, *datasets* e características utilizadas.

Unsupervised Grammar Induction in Natural Language Processing: A Systematic Mapping Study

AUTHOR1*, Institution, Country
AUTHOR2, Institution, Country

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

ACM Reference Format:
Author1 and Author2. 2018. Unsupervised Grammar Induction in Natural Language Processing: A Systematic Mapping Study. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2018), 37 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The human language is a prominent feature of human intelligence, with grammar playing an essential role in communication. Grammar can be applied in computers in two ways: through predefined structure rules or induced structure. ELIZA was one of the first Natural Language Processing (NLP) applications that used predefined structure rules to simulate a conversation [146]. The latter approach tries to discover syntax structures from data. Discovering syntax structures could be helpful in many applications beyond computer science, such as Bioinformatics and Psycholinguistics. In Bioinformatics, DNA sequences are treated as text that aims to identify critical genetic structures. Meanwhile, the capacity to extract structures could be applied in psychology to develop language acquisition models.

The task of discovering syntax structure incorporates a range of, including grammar acquisition, unsupervised parsing, grammatical inference, or grammar induction. The use of those terms depends on the specific context in which that task is applied. In formal languages, grammatical inference is conventionally employed. In research applied to Psycholinguistics or linguistics, grammar acquisition holds greater prevalence. Within the domain of computer science, particularly in the field of NLP, grammar induction is usually used, and with a focus on machine learning, unsupervised parsing is frequently used.

In NLP applications, grammar induction proves useful for various tasks, including grammar correction, information extraction, and text simplification, to name a few. Grammar induction can be approached in an unsupervised way (UGI), a semi-supervised way (SSGI), or a supervised way (SGI). SGI demonstrated remarkable efficacy in many works, achieving accuracy rates exceeding 95% [80], their unsupervised counterparts present a considerable challenge, often falling short of this benchmark. UGI stands out as one of the most challenging tasks in computer science, and several reasons contribute to its inherent difficulty.

* Authors' addresses: Author1, email@email, Institution, Address, City, State, Country, postcode; Author2, Institution, Address, City, Country, email@email.

2476-1249/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

First, natural language can be viewed as a dynamic system, as its syntactic and semantic structures vary according to social interactions over time [118]. For example, the English used by William Shakespeare differs from contemporary English. Second, the complexity of syntactic rules across different languages presents a significant challenge, as there is no consensus among linguists regarding the grammatical rules for specific structures [54]. Third, understanding and applying the language acquisition process to computers is a challenge. Finally, generating optimal parse trees is an NP-complete problem [116].

With the emergence of transformer-based models [145], grammar induction has gained more attention. Some advocates suggest that Pre-trained Language Models based on transformers understand syntax [7]. However, they still exhibit lower accuracy in UGI than other approaches [68]. Traditional approaches in UGI include the Expectation-Maximization (EM) algorithm [34] and Bayesian Inference, and more recently, in neural networks. Nevertheless, research in UGI lacks a clearly defined trajectory, unlike more established subfields in NLP, like machine translation.

Consequently, many works attempt to improve accuracy by employing an exhaustive exploration of strategies, even though UGI is active research in this field that spans more than two decades. Some of these strategies are confined to specific studies. The diversity of strategies poses an additional challenge in conducting comprehensive research within this domain.

To tackle these issues, we propose a systematic mapping to organize the area regarding what has been done and how it was accomplished. Our study pursues two primary objectives. Firstly, we aim to evaluate the effectiveness of different techniques in grammar induction, particularly those focused on raw text, across various languages and corpora. Secondly, we aim to identify and document the difficulties and gaps encountered in the grammar induction task.

The rest of the article is structured as follows: In Section 2, we present background on grammar and grammar induction. Section 3 presents related works. Section 4 describes the systematic mapping review. Section 5 reports threats to validity. Section 6 presents the results. Section 7 discusses the findings, and Section 8 concludes the article.

2 BACKGROUND

In this chapter, the necessary concepts for the understanding of this work are presented.

2.1 Grammar

The word “grammar” originates from the Greek phrase $\gamma\rho\alpha\mu\mu\alpha\tau\iota\kappa\eta\ \tau\epsilon\chi\upsilon\eta$, which means “grammatical art” [93]. This expression was used to refer to the discipline focused on the art of writing literature. Grammar is also synonymous with language theory. Among the most prominent language theories are Generative grammar [25] and Cognitive grammar [76] [54], representing rationalist and empiricist approaches, respectively [31].

Supporters of the rationalist view, exemplified by Chomsky, argue that sentences can be generated through rules that operate on a set of symbols, which are combined, rearranged, and transformed [22]. Chomsky’s theory delineates language into two skills: competence and performance. Competence refers to the knowledge of the language held by the speaker and the listener. Performance refers to the capacity of both the speaker and the listener to use the language, whether in generation or comprehension [25]. According to Chomsky, performance and competence are innate, as he believes that predefined linguistic knowledge is present at birth. He refers to this as the *"language acquisition device"* [24]. Although linguistic performance is widely accepted in academia as an innate ability, the idea of innate competence is contested by cognitive scientists [139]. This debate is essential for the UGI task since understanding how children acquire language could be a great promise for advancing UGI.

In contrast to generative grammar, cognitive grammar disregards Chomsky’s concept of grammar rules and is formalized through Context-Free Grammar (CFG). According to Lakoff [75], language is constituted by

constructions that arise not from predetermined rules but based on usage and empirical evidence. It is important to note that cognitive grammar [76] is different from cognitive linguistics [75]. Despite their simultaneous emergence, they encompass distinct definitions. The former refers to a theory of language, while the latter constitutes a research program that includes cognitive grammar [95].

From this perspective, a child does not acquire correct syntactic constructions by applying rules from a *"language acquisition device"*. Instead, they learn how to use them based on usage (whether passive or active). Chomsky, however, contests this viewpoint, arguing that the child does not receive sufficient stimuli to comprehend syntactic rules [10].

Generative and cognitive theorists define grammar primarily in terms of phonology, syntax, and semantics [25] [40]. Generative grammar predominantly emphasizes syntax, whereas cognitive grammar places greater emphasis on semantics in sentence structure [54]. Applied linguists extend the grammar scope to include morphology as well [1]. In the context of UGI research, a wide spectrum of elements, ranging from phonology to pragmatics, are considered in exploring linguistic structures. UGI uses two kinds of grammar representation to discover structures in texts: constituency grammar and dependency grammar.

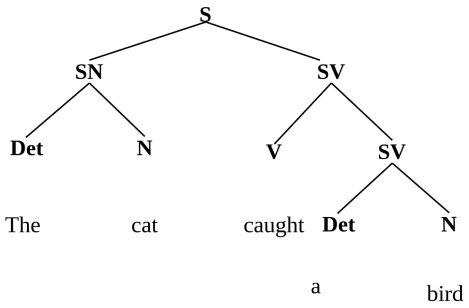


Fig. 1. Constituency grammar syntax tree

Constituency grammar, also known as phrase structure grammar, is closely related to generative grammar, as it was formalized by Chomsky using CFG [22]. Although CFG is designed for constituency grammar, generating the constituent parse tree is not restricted to using CFG. Other formalisms, such as Combinatory Categorial Grammar (CCG) [133], Definite-Clause Grammar [39], and Lexical Functional Grammar [18], can also be employed for this purpose. The syntactic tree used in constituency grammar can be constructed in the same way as in formal languages (see Figure 1): top-down or bottom-up, and it uses the same formalism, including terminals, non-terminals, and production rules.

Preceding the seminal publication of Chomsky's renowned work, *Syntactic Structures* [23], which subsequently led a surge of critical discourse on his theoretical framework [54], Tesnière was already working on a novel syntactic formalism. However, it was not published before his death [138]. Tesnière is widely acknowledged for pioneering the initial efforts to systematize dependency grammar [33]. Although Tesnière did not explicitly propose a theory of cognitive grammar, many elements of his formalism are similar to cognitive grammar proposed by Langacker [77]. It is essential to underscore that cognitive grammar formed the foundation for developing dependency grammar.

Constituency grammar is based on the structure and construction of sentences. In contrast, dependency grammar addresses sentence construction and considers the roles of the relations between words. In dependency grammar, parse trees are constructed based on the connections between the governing head term and its dependent terms within a sentence. In Figure 2, the arrows connect the head term (origination point of the arrow) and

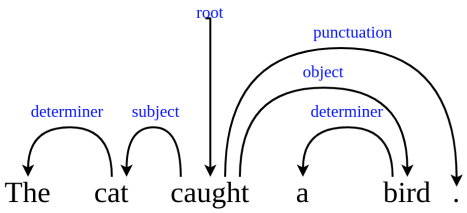


Fig. 2. Dependency grammar syntax tree

the dependent term (terminus of the arrow). The labels positioned above the arrows denote the specific type of linguistic relationship.

Dependency grammar diverges from constituency grammar in several aspects. First, dependency grammar lacks ternary relationships, a notable departure from constituency grammar, where more than two constituents can descend from the same previous constituent [33]. Second, dependency grammar emphasizes semantics in syntactic functions between relations, providing more rich information [32]. Third, constituent trees have the potential for exponential growth regarding node count, a characteristic not shared by dependency trees [61] [125]. Fourth, parsing an optimal constituency grammar tree is NP-complete [116], whereas dependency grammars can be polynomially under the condition of projectivity (permitting intersection between dependency arcs) [74]. Finally, dependency grammar allows the incorporation of aspects from the constituency grammar. For example, in dependency grammar, terminals may be represented as words lacking dependents, while words with dependents can represent non-terminal constituents.

Various representations of dependency grammar exist, with notable examples including Word Grammar [95, p. 526], Meaning–Text Theory [94], and Universal Dependency (UD). In each dependency grammar representation,a token may have only one dependent, yet that same token may, in turn, be dependent on multiple others. Every token, except for the ROOT, must rely on another token. Some studies posit that the incoming arc should denote the head of the relation, while alternative perspectives consider that it must denote dependency [67]. In this work, we adopt the direction proposed in the framework of UD, which defines the arrow of the arc as originating from the head term.

2.2 Grammar Induction

UGI aims to induce syntax rather than part-of-speech (POS) tags. Induce POS tags are typically employed to reduce data sparsity and computational complexity [71] [57] [98]. Few UGI works do not use morphosyntactic or any additional information. We refer to this type of UGI as “strong UGI” to distinguish them.

Within the domain of UGI, methodologies applied to constituent and dependency grammars are broadly categorized into generative, discriminative, and heuristic models. Generative models model the joint probability $P(x, y)$ between the sentence x and the parsing tree y. The model is referred to as generative because it generates examples from the probability distribution of x. In UGI, the generative model aims to produce instances of syntax trees representing a given sentence. Conversely, discriminative models characterize the conditional probability $P(y|x)$, which calculates the likelihood of parsing tree y occurring based on the sentence x.

The generative approach can be categorized into three groups: structural, non-parametric, and parametric [141]. The structural generative approach, also known as structural search or structural learning, does not impose a fixed grammar on the model. Instead, it is acquired, along with the model parameters, through heuristic methods. Among the main techniques are the Hidden Markov Model (HMM) [135], and the Latent Tree Graphical Model [98], to name a few.

Parameterized models have a fixed number of parameters. However, it is important to note that asserting that non-parametric models lack parameters would be inaccurate. These models have an indeterminate number of parameters, potentially approaching infinity as the dataset expands [45]. This approach allows the model to adjust to its complexity and the amount of data.

Ultimately, the parameterized approach is the most widely used in UGI. This approach begins with a fixed grammar model, and then the model parameters are optimized to find the best parse tree for a given sentence. The main techniques used in parameter optimization include Bayesian inference, the Inside-Outside algorithm (IO) [8], the Expectation-Maximization algorithm (EM) [34], and neural networks (including language models).

3 RELATED WORK

The varied terminologies and the diversity of areas that explore GI contribute to the publication of surveys across diverse fields. Surveys have been conducted in software engineering [134], formal languages [11] [42], or different training methods [43]. In NLP, four surveys and one systematic review related to UGI have been published since 2002, but none are mapping studies.

In 2010, Sankaran [111] conducted an extensive survey. The author focused on different strategies used in UGI across one and multiple languages. The study revealed that the majority of models use the EM algorithm. However, this work does not include discriminative approach studies. In contrast, generative and discriminate approaches in UGI were investigated by Han et al. [51]. Nevertheless, their investigation did not include constituency grammar. Providing a comprehensive overview of UGI, D'Ulizia et al. [36] centered their work on techniques rather than specific studies, including thoroughly exploring supervised methods. Additionally, [89] presents a survey about the dependency grammar research, spanning from Klein and Manning's seminal work in 2004 [72] to 2016. Their work provides a panoramic view of the field and compiles the research across various languages.

The only systematic review was published by Muralidaran et al. [96]. The authors examined 43 publications published between 2000 and 2020 from 198 studies. These studies were categorized according to nine criteria, including study type, theoretical underpinning of grammar, grammar representation, and employed features. The search strategy adopts a more specific approach that targets particular models and studies. Furthermore, they included formal language studies.

Our Systematic Mapping Study focuses on unsupervised grammar induction applied only to NLP. Unlike the approach taken by [96], we deliberately omitted theoretical studies in our review. We aim to organize, as comprehensively as possible, all techniques, resources, and methodologies applied to the task of UGI.

4 SYSTEMATIC MAPPING PROCESS

Systematic mapping study (SMS) aims to construct a classification map for topics studied in a field of interest. It is used to know what topics have been covered in literature and which methods and techniques are used [101]. Unlike a Systematic literature review (SLR), SMS tries to understand the area to provide a broad overview of the topic. Unlike SMS, SLR covers a smaller and specific range of studies [70].

This survey follows the guidelines proposed by [70] and [101]. In Section 4.1, we plan the mapping execution and present the research questions. In Section 4.5, we execute the planning by selecting studies and extracting data. Finally, in Section 6, we present the synthesis of the results.

4.1 Planning

This mapping defines six research questions in Section 4.2. The choice of *strings* for searches, as described in Section 4.3, was defined empirically, aiming to group the most significant number of studies in the area. The searches (Section 4.3) were conducted in the following databases: IEEE, ACM, Scopus, Google Scholar, Springer,

ACL. Table 1 shows the number of studies per database. The numbers were obtained in March of the year 2023. We used three selection criteria (Section 4.4): inclusion, exclusion, and quality with 4, 9, and 3 criteria, respectively.

Table 1. Overview of databases

Database	Items	Link
ACL	83K	https://aclanthology.org
IEEE	5922K	https://ieeexplore.ieee.org
ACM	691K	https://dl.acm.org
Scopus	87000K	https://www.scopus.com
Google Scholar*	389000K	https://scholar.google.com.br
Springer	15753K	https://link.springer.com

*[50]

4.2 Research Questions

The research questions focus on exploring the characteristics, techniques, and methods presented in the literature, considering the various terminologies employed in natural language processing. The following research questions are defined:

RQ1 *What are the influential works of the field?* By answering this RQ, we pretend to have a view of the influence of the field of UGI over time.

RQ2 *What approaches have been used in UGI?* By answering this RQ, we pretend to construct a map of techniques and models used in different grammar representations. That will help better understand why those techniques and models were used.

RQ3 *What resources and methodology are applied in the studies?* By answering this RQ, we pretend to organize the resources and methodology applied in the studies. It includes identifying typical training constraints and the corpus most used.

RQ4 *Does language influence the results of different approaches?* By answering this RQ, we pretend to determine if the differences across languages can significantly influence results.

RQ5 *What evaluation metrics are used?* By answering this RQ, we organize the metrics used in UGI.[**talvez tirar?**]

RQ6 *What are the trends in the field?* By addressing this research question, we aim to uncover the evolving patterns and tendencies within this domain over time

4.3 Search strategy

As mentioned in Section 1, the task of inducing grammar from text is associated with various nomenclatures. Furthermore, several studies employ the terms GI and UGI. The search strings presented in this work were meticulously designed to include all those variations to recover as many studies as possible applied to NLP. Search strings were constructed based on the following terms:

- **unsupervised**: Include unsupervised training studies.
- **dependency**: Some studies may rely only on constituency grammar.
- **constituency**: Some studies may rely only on dependency grammar.
- **parsing**: Some important studies may employ this term instead of “grammar induction”.
- **grammar**: Exclude studies that apply UGI in contexts unrelated to NLP.
- **induction**: It is an important keyword for the task of UGI.

- **latent**: Some studies may use that term instead of use “parsing” or “grammar induction”.
- **tree**: This term often appears in conjunction with “latent" and within expressions like “tree structure".
- **grammatical**: This term pertains to the expression "grammatical inference".
- **inference**: This term pertains to the expression “grammatical inference".
- **syntactic**: Some computational linguistics works may employ this more formal term for UGI.
- **structure**: Usually used in conjunction with “syntactic" and “tree."

At the outset, we chose a basic string search, further refined through an empirical approach. In formulating search strings, it is advisable to incorporate pluralized terms and synonyms [101]. Several terms, including dependencies, constituencies, trees, and structures, were utilized in their plural and synonymous forms. However, these variations did not yield significant disparities in the quantity of studies retrieved through the search. The terms “syntactic” and “structure” were omitted from the final search string due to potential noise and the complexity increase of the string. The refined search string is as follows:

*“unsupervised parsing” **OR** “grammar induction” **OR** “latent tree induction” **OR** ((“dependency parsing” **OR** “constituency parsing” **OR** “grammatical inference”) **AND** (unsupervised **OR** induction))*

Uniform search string conventions are not consistent across various databases. Scopus, Springer, and Google Scholar use the same format, whereas IEEE and ACM employ distinct variants. The meticulously formatted search strings applied in our selection process are presented as follows:

- **IEEE** “Publication Title": “unsupervised parsing" **OR** “Abstract": “unsupervised parsing" **OR** “Author Keywords": “unsupervised parsing" **OR** “Publication Title": “grammar induction" **OR** “Abstract": “grammar induction" **OR** “Author Keywords": “grammar induction" **OR** “Publication Title": “latent tree induction" **OR** “Abstract": “latent tree induction" **OR** “Author Keywords": “latent tree induction" **OR** ((“Publication Title": “dependency parsing" **OR** “Abstract": “dependency parsing" **OR** “Author Keywords": “dependency parsing" **OR** “Publication Title": “constituency parsing" **OR** “Abstract": “constituency parsing" **OR** “Author Keywords": “constituency parsing" **OR** “Publication Title": “grammatical inference" **OR** “Abstract": “grammatical inference" **OR** “Author Keywords": “grammatical inference") **AND** (“Publication Title": “induction" **OR** “Abstract": “induction" **OR** “Author Keywords": "induction" **OR** “Publication Title": “unsupervised" **OR** “Abstract": “unsupervised" **OR** “Author Keywords": “unsupervised"))

- **ACM** Title: (“unsupervised parsing") **OR** Abstract: (“unsupervised parsing") **OR** Keywords: (“unsupervised parsing") **OR** Title: (“grammar induction") **OR** Abstract: (“grammar induction") **OR** Keywords: (“grammar induction") **OR** Title: (“latent tree induction") **OR** Abstract: (“latent tree induction") **OR** Keywords: (“latent tree induction") **OR** ((Title: (“dependency parsing") **OR** Abstract: (“dependency parsing") **OR** Keywords: (“dependency parsing") **OR** Title: (“constituency parsing") **OR** Abstract: (“constituency parsing") **OR** Keywords: (“constituency parsing") **OR** Title: (“grammatical inference") **OR** Abstract: (“grammatical inference") **OR** Keywords: (“grammatical inference")) **AND** (Title: (“unsupervised") **OR** Abstract: (“unsupervised") **OR** Keywords: (“unsupervised") **OR** Title: (“induction") **OR** Abstract: (“induction") **OR** Keywords: (“induction")))

- **Scopus, Springer, Google Scholar e ACL** “unsupervised parsing" **OR** “grammar induction" **OR** “latent tree induction" **OR** ((“dependency parsing" **OR** “constituency parsing" **OR** “grammatical inference") **AND** (unsupervised **OR** induction))

4.4 Criteria Selection

In the study selection process, it is necessary to clearly define which criteria the study is eligible for either inclusion or exclusion. In cases where the selected studies meet all the established criteria yet still exhibit a considerable volume of studies to analyze, applying quality assessment is recommended [101]. This selection process was executed interactively. Criteria were removed, included, modified, or joined with two or more criteria until they achieved a good enough list of criteria presented in Table 2.

We have employed rigorous quality criteria for this mapping endeavor, excluding studies characterized by poor accuracy, redundancy, reliance on unconventional resources, or insufficient methodological exposition.

Table 2. Selection criteria applied

Criteria	N	Description
Inclusion	I1	Include studies written in English.
	I2	Include primary studies publications as papers, journals, or workshops (peer-reviewed or not).
	I3	Include experimentation in the study.
	I4	Include unsupervised training.
Exclusion	E1	Exclude publications before 1992.
	E2	Exclude studies not related directly to NLP.
	E3	Exclude projects, reports, thesis, books, posters, or any gray literature or work (or compiled of works) that describes a published study.
	E4	Exclude short papers or unpublished studies.
	E5	Exclude studies that reproduce methods, or are similar or extensions to a more relevant study, to only one language or use artificial corpus.
	E6	Exclude studies that do not use unsupervised training OR apply transfer learning.
	E7	Exclude comparative and reproduction studies.
	E8	Exclude secondary, theoretical studies or not do experiments to induce grammar.
	E9	Exclude studies not written in English.
Quality	Q1	Reject studies that use unusual corpus (applied only to studies that do not use language models).
	Q2	Reject studies that present poor results (considering the year of publication) or do not present enough results and methodology information.
	Q3	Reject studies that are not focused on grammar induction to NLP
	Q4	Reject studies that are similar to a more significant study.

4.5 Execution

In this section, we execute the planning presented in the previous section. We searched publications and applied criteria to select studies.

4.6 Search string results

There is no standardization of how searches can be performed in different databases. Google Scholar and Springer do not support the search for keywords and abstracts. They only search across the entire document, including

references. Furthermore, the search engine in the ACL repository only displays the initial 100 studies in the results. Additionally, we conducted searches [1] in the ACL repository using the BibTeX provided by the portal [2]. Of the 83,369 publications included in this search, 53.1% did not include abstracts. Only the IEEE, Scopus, and ACM can search by title, abstract, and keywords among the databases selected for this mapping. Therefore, we exclusively utilized these databases to evaluate the different search strings.

The vast database and exhaustive document search present a challenge for effective searching in Google Scholar. Using the final search string, the search engine retrieved 18000 studies. However, Google Scholar only displays the top 1000 most relevant studies (it is not clear what is relevant to Google Scholar). To maximize the inclusion of studies, we divided the research into six segments, as detailed in Table 3.

Table 3. Retrieved studies on Google Scholar without use stop criteria.

Interval	**Studies**
1992 - 2007	3910
2008 - 2013	4210
2014 - 2018	4350
2019 - 2022	3840
2023 -	252

This strategy increases the likelihood of finding important studies. However, it comes at the cost of yielding many irrelevant ones. In this situation, it is recommended to establish a stopping criteria [101]. We defined criterion studies that do not present any terms described in Section 4.3 in five consecutive studies displayed in the Google Scholar search engine. This number was defined empirically after searching with several different values. Only for 2023 were all studies selected without considering this stopping criterion. With this stopping criterion, we reduced the scope from 18 thousand studies to only 559. Table 4 presents the results of the studies retrieved in each database.

Table 4. Retrieved studies with duplicates

Database	**Studies**
Scopus	749
ACM	119
IEEE	82
Springer	2501
Google Scholar	559
ACL	198
All	4281

We annotated each study's source of origin upon retrieval. Regarding duplicates, priority was assigned in the following order: SCOPUS ⇒ ACM ⇒ IEEE ⇒ Springer ⇒ Google Scholar ⇒ ACL. Following the removal of duplicates, the ultimate count of studies is detailed in Table 5

[1] [github code]
[2] https://aclanthology.org/anthology+abstracts.bib.gz

Table 5. Retrieved studies without duplicates

Database	Studies
Scopus	714
ACM	23
IEEE	3
Springer	2336
Google Scholar	175
ACL	10
All	3259

4.7 Selection process

This process was initiated in March 2023 and involved four iterative steps and a quality assessment. In the initial step, exclusion criteria one through four were applied concerning an essential examination of titles, abstracts, and metadata. Most studies analyzed in this step were unrelated to grammar induction, with a significant portion (83%) from the Springer database. Following this initial step, 1,533 studies remained, constituting 47.1% of the total. Since more information was required before making inclusion decisions, no studies were incorporated into the final selection at this stage. However, we identified a subset of high-quality studies that advanced directly to the final phase.

In the second step, all remaining studies that did not advance to the final stage were re-evaluated. We scan the full text when necessary. All exclusion criteria were rigorously applied. Similar to the initial step, some studies met the criteria for quality assessment and were not re-evaluated in the subsequent phase. This step aimed to spot studies that exhibited strong potential for inclusion in the ultimate selection of studies. At the end of this round, 261 studies were selected (8% of the total number of studies).

In the third step, studies were examined in full when necessary, focusing on the introduction to apply exclusion criteria. This step aimed to identify selection failures in the previous steps. This step resulted in 126 studies.

Finally, in the final step, all remaining studies were read fully. All exclusion criteria were once again applied. After that, quality assessment criteria were used for all 90 remaining studies. Forty-nine studies compounded the final group of selected studies. After duplicate removal, the selection process is presented in Figure 5, and the list of select studies is presented in Table 6.

Figure 3 illustrates the word cloud generated from the titles of the 3259 studies obtained using the final search string outlined in Section 4.3.

Figure 4 presents a word cloud generated from the abstracts of the 49 selected studies. In contrast to Figure 3, most words are not present in the terms introduced in Section 4.1. Both word clouds in Figures 3 and 4 consist only of nouns, adjectives, and proper names. The original tokens were considered in the construction of the cloud, except for pluralized nouns and proper names, which were singularized. Among the most frequently occurring single words in the abstracts, we find *model* in 42 studies, *unsupervised* in 40 studies, *grammar* in 34 studies, *dependency* in 34 studies, *parsing* in 33 studies, *induction* in 26 studies, and *state-of-the-art* in 24 studies.

Fig. 3. Word built with the titles of 3259 studies retrieved in this mapping review.

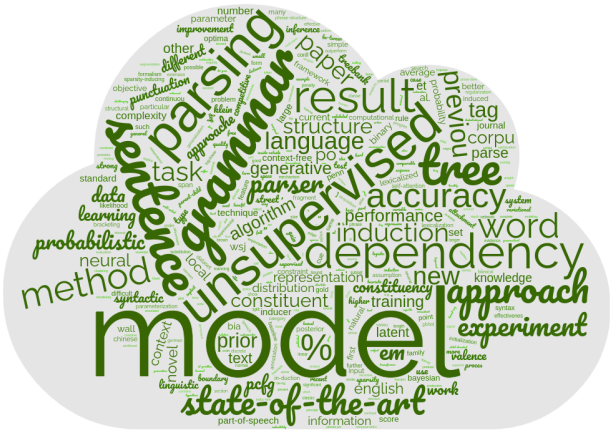


Fig. 4. Word cloud with all *abstracts* from 49 studies

Table 6. Selected studies

ID	Title	Ref	Code
S1	A Feature-Rich Constituent Context Model for Grammar Induction	[47]	Not available online
Continue in the next page			

Table 6 – Selected studies

S2	A Generative Constituent-Context Model for Improved Grammar Induction	[71]	https://github.com/davidswelt/dmvccm/tree/master/lq-dmvccm/dmvccm [3]
S3	An All-Subtrees Approach to Unsupervised Parsing	[16]	Not available online
S4	Annealing Structural Bias in Multilingual Weighted Grammar Induction	[121]	Not available online
S5	Are pre-trainded language models aware of phrases? simple but strong baselines for grammar induction	[68]	https://github.com/galsang/trees_from _transformers
S6	Breaking Out of Local Optima with Count Transforms and Model Recombination- A Study in Grammar Induction	[130]	Not available online
S7	Capitalization cues improve dependency grammar induction	[128]	Not available online
S8	Compound Probabilistic Context-Free Grammars for Grammar Induction	[69]	https://github.com/harvardnlp/compound-pcfg
S9	Corpus-Based Induction of Syntactic Structure- Models of Dependency and Constituency	[72]	https://github.com/davidswelt/ dmvccm/tree/master/lq-dmvccm/dmvccm [4]
S10	Dependency Grammar Induction with Neural Lexicalization and Big Training Data	[52]	https://github.com/LouChao98/neural_based_dmv
S11	Enhancing Unsupervised Generative Dependency Parser with Contextual Information	[53]	https://github.com/LouChao98/neural_based_dmv
S12	Exploiting Reducibility in Unsupervised Dependency Parsing	[92]	Not available online
S13	Fast Unsupervised Incremental Parsing	[113]	Not available online
S14	fast-r2d2: a pretrained recursive neural network based on pruned cky for grammar induction and text representation	[59]	https://github.com/alipay/StructuredLM_RTDT
S15	From baby steps to leapfrog: How “less is more" in unsupervised dependency parsing	[125]	Not available online
S16	Gibbs Sampling with Treeness Constraint in Unsupervised Dependency Parsing	[91]	Not available online
S17	Guiding Unsupervised Grammar Induction Using Contrastive Estimation	[120]	Not available online
S18	Identifying Patterns for Unsupervised Grammar Induction	[112]	Not available online
S19	Improving Unsupervised Dependency Parsing with Richer Contexts and Smoothing	[57]	Not available online
S20	Inducing Tree-Substitution Grammars	[29]	Not available online
S21	Lateen EM: Unsupervised training with multiple objectives, applied to dependency grammar induction	[126]	Not available online
	Continue in the next page		

[3]Unofficial - Implementation by Franco M. Luque
[4]Unofficial -Implementation by Franco M. Luque with different initialization

Table 6 – Selected studies

S22	Logistic Normal Priors for Unsupervised Probabilistic Grammar Induction	[27]	Not available online
S23	Neural Bi-Lexicalized PCFG Induction	[151]	https://github.com/sustcsonglin/TN-PCFG
S24	PCFGs can do better: Inducing probabilistic context-free grammars with many symbols	[152]	https://github.com/sustcsonglin/TN-PCFG
S25	Punctuation: Making a point in unsupervised dependency parsing	[127]	Not available online
S26	Second-Order Unsupervised Neural Dependency Parsing	[150]	https://github.com/sustcsonglin/second-order-neural-dmv
S27	Shared Logistic Normal Distributions for Soft Parameter Tying in Unsupervised Grammar Induction	[28]	Not available online
S28	Simple Robust Grammar Induction with Combinatory Categorial Grammars	[12]	Not available online
S29	Simple Unsupervised Grammar Induction from Raw Text with Cascaded Finite State Models	[104]	Not available online
S30	Sparsity in Dependency Grammar Induction	[46]	Not available online
S31	Spectral Unsupervised Parsing with Additive Tree Metrics	[98]	Not available online
S32	Stop-probability estimates computed on a large corpus improve Unsupervised Dependency Parsing	[90]	Not available online
S33	Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling	[115]	https://bit.ly/gitlab_structformer
S34	The Return of Lexical Dependencies- Neural Lexicalized PCFGs	[154]	https://github.com/neulab/neural-lpcfg
S35	Three Dependency-and-Boundary Models for Grammar Induction	[129]	Not available online
S36	Unsupervised and few-shot parsing from pretrained language models	[153]	Not available online
S37	Unsupervised Dependency Parsing with Acoustic Cues	[99]	https://github.com/jpate/predictabilityParsing
S38	Unsupervised Dependency Parsing without Gold POS tags	[124]	Not available online
S39	Unsupervised Dependency Parsing- Let's Use Supervised Parsers	[79]	Not available online
S40	Unsupervised Grammar Induction with Depth-bounded PCFG	[65]	https://github.com/lifengjin/db-pcfg
S41	Unsupervised Induction of Tree Substitution Grammars for Dependency Parsing	[14]	Not available online
S42	Unsupervised Latent Tree Induction with Deep Inside-Outside Recursive Autoencoders	[5]	https://github.com/iesl/diora
S43	Unsupervised Learning of PCFGs with Normalizing Flow	[66]	https://github.com/lifengjin/acl_flow

Continue in the next page

Table 6 – Selected studies

S44	Unsupervised Learning of Syntactic Structure with Invertible Neural Projections	[56]	https://github.com/jxhe/struct-learning-with-flow
S45	Unsupervised Neural Dependency Parsing	[64]	Not available online
S46	Unsupervised Parsing via Constituency Tests	[20]	https://github.com/harvardnlp/urnng
S47	Unsupervised parsing with U-DOP	[17]	Not available online
S48	Using left corner parsing to encode universal structural constraints in grammar induction	[97]	Not available online
S49	Viterbi Training Improves Unsupervised Dependency Parsing	[131]	Not available online

5 THREATS TO VALIDITY

In this section, we discuss the validity threats in this study and present actions to mitigate them. We follow [101] guidelines that describe threat validity into four groups: descriptive, theoretical, generalizability, interpretive, and repeatability. They are discussed as follows:

Study search. As previously mentioned, the field of grammar induction employs various terminologies, posing a significant threat to missing relevant studies while including unrelated studies. Furthermore, some databases, such as Google Scholar and Springer, may retrieve more studies than are feasible for us to analyze. Limiting the search without missing important work is a persistent challenge. To address that threat, we divide the search in Google Scholar into four segments and employ an empirical process for string search selection.

Furthermore, our search extended to diverse databases, including a specialized computer science repository and a specialized NLP repository. We employed snowball sampling techniques outlined by Jalali and Wohlin [62] to account for any potentially missed works. Specifically, we used backward and forward snowball sampling in one level using as seed the ten most cited studies selected in the fourth step, according to Google Scholar. However, we perform only E1, E2, E3, and Q4 criteria. We did not include these studies in the final group of select studies or the analysis. Instead, we list the top 15 most relevant studies based solely on citation counts for future reference [link omitted].

Additionally, the exact string search could retrieve different studies on different moments. To address this threat, we execute the final string search three times (once a day for three consecutive days) and take only studies that appear in all searches.

Study selection. Researcher bias may threaten validity during data selection and extraction. To mitigate that threat, we separate the revision and selection of the study among the authors of this study. Furthermore, the selection process and quality assessment are broken down into four steps. Each study could be reviewed four times in different moments and in different ways when necessary. We define the list of criteria as objectively as possible based on the goals of this work. The criteria list is constructed with a strong emphasis on objectivity, aligning with the study's objectives. The quality assessment was applied after the four steps were concluded. Furthermore, the criteria for inclusion remain consistent with the initial parameters set at the beginning of this mapping. However, slight adjustments have been made to both the exclusion criteria and the criteria used for quality assessment from their original configurations. Moreover, we provide a list of all documents retrieved after duplicate removals [link omitted].

The categorization of the studies raises a potential concern. To mitigate that, we take some actions. In response, we adopted a more stringent approach in classifying studies as unsupervised. We have established that any work relying on manually crafted linguistic rules or self-training falls into semi-supervised work. Consequently, such

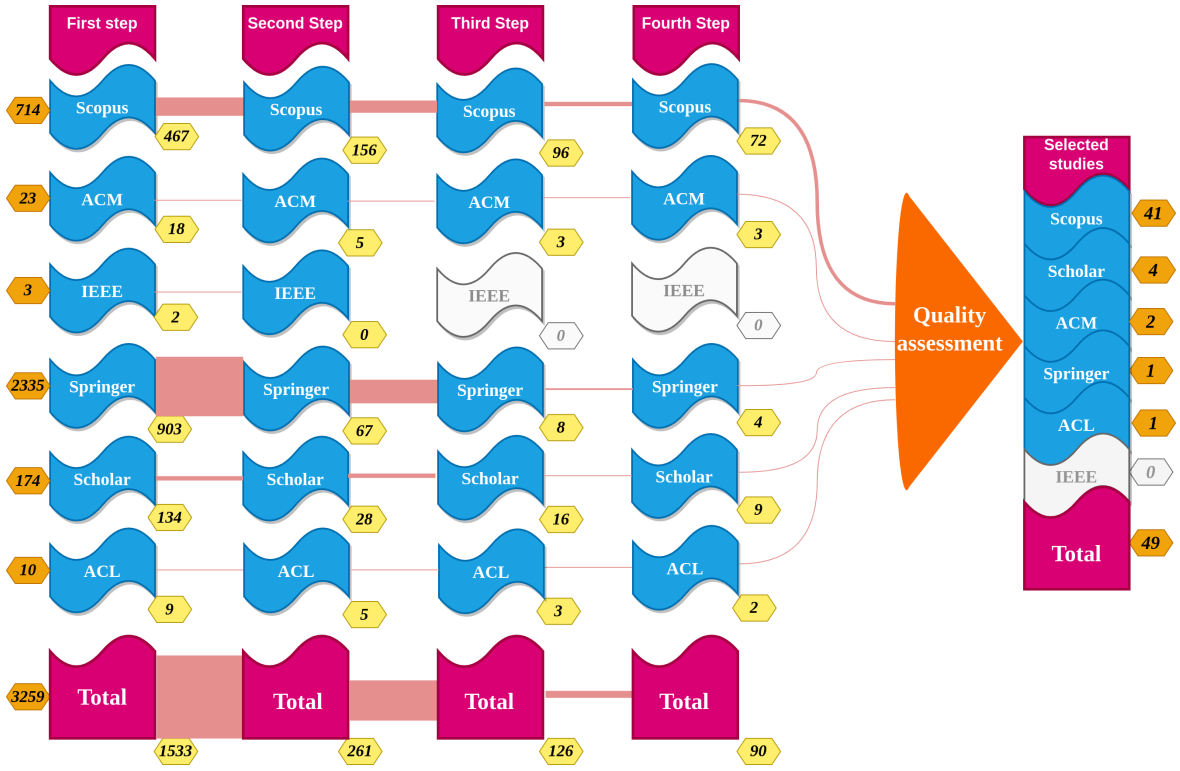


Fig. 5. Selection process

works were excluded from the selection criteria. Another decision revolved around including S13, which does not explicitly utilize dependency or constituency grammar. Ultimately, we opted to include S13 due to its frequent comparison to constituency models across numerous studies, offering valuable insights in this research domain. Some studies use heuristics, but we decided to include only those that do not use generative or discriminative approaches in the heuristic group.

Another challenge arises from potential gaps in information within the databases, including instances where author names and DOIs are absent. This issue is addressed through manual verification of each study lacking essential details, ensuring completeness and reliability.

Data extraction One research author conducted revisions, while the other focused on data extraction, following Kitchenham et al. [70] guidelines. The absence of information in certain studies and reliance on manual data extraction potentially threaten validity. To mitigate that, for some studies, we validate the data by analyzing the reproduction of the study by other studies using the same methodologies. Furthermore, most of our guidelines for data extraction are objective. The studies that we could not identify the data were discarded in the quality assessment step.

Research validity. Since there is no well-defined classification of methods used in grammar induction, some studies could be misclassified and fall in threat validity. To minimize this threat, we use the classical classification in machine learning (discriminative and generative) and divide them into subgroups influenced by Sankaran et al. [111].

6 RESULTS

In the following section, we present a structured framework comprising seven Sections. The primary focus of the initial six sections centers on the comprehensive exploration of the six research questions outlined in Section **??**. The final Section is designated for discussion. Details regarding paper citations in the subsequent sections were extracted from *Google Scholar* on June 6, 2023.

6.1 RQ1 - What are the influential works of the field?

In general, influential works are often characterized by many citations. However, our objective is not to present the most cited works exclusively but to gain insight into the evolution of influential works over time. Figure 6 visually represents these influential works, with citations serving as the primary metric. The orange bars correspond to studies on constituency grammar, the pink bar represents studies on dependency grammar, and the green bar indicates studies that include both grammatical approaches. On the left side of the figure, we observe the most highly cited study S9. Significantly, the study S36 chosen for examination, positioned on the far right, had not garnered any citations until June 6, 2023. It is worth mentioning that Klein and Manning contributed to both the first and second most cited studies.

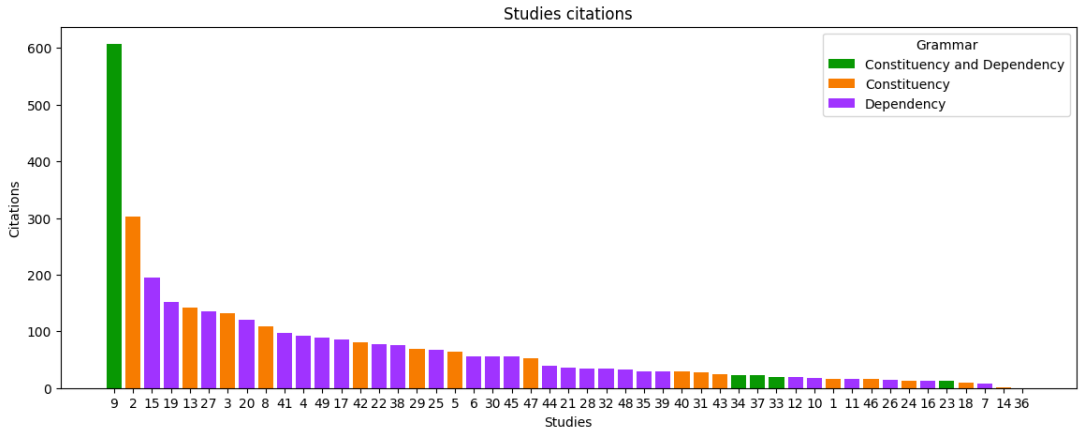


Fig. 6. Most influential works based on citations

Beyond understanding influence in a broader context, it is crucial to discern a work's influence within its peer community. We have constructed a co-authorship network with all selected studies to achieve this. If study X references study Y or is referenced by study Y, an edge will be established between X and Y. Tu analyze the networks, we use centrality metrics. Centrality is a metric for measuring impact in co-authorship networks [149]. Following Erjia et al. [149], we consider four central metrics: betweenness, closeness, degree, and PageRank. The results are presented in Table 7. S9 stands out as the most central study among the selected ones. However, despite its centrality, it does not rank among the top ten studies regarding degree centrality. That can be attributed to its status as the oldest study in the selection, and it does not cite any other studies in this mapping review. Figure 7 illustrates the co-authorship network constructed using betweenness centrality. The colors denote the betweenness centrality score, transitioning from purple to red across the RGB spectrum to signify the highest to lowest scores, respectively.

	degree_centrality		closeness_centrality		betweenness_centrality		pagerank	
1	15	0,89796	**9**	0,90741	**9**	0,1884	**9**	0,05875
2	3	0,63265	19	0,73134	2	0,09421	19	0,04155
3	27	0,57143	6	0,7	8	0,07391	2	0,03879
4	19	0,55102	2	0,69014	19	0,05955	6	0,03764
5	13	0,4898	8	0,66216	6	0,05125	8	0,03485
6	29	0,42857	27	0,62821	15	0,02527	15	0,02871
7	49	0,40816	22	0,62025	27	0,02121	27	0,02735
8	43	0,38776	15	0,6125	13	0,01958	22	0,02606
9	44	0,36735	13	0,6125	22	0,019	13	0,02522
10	12	0,36735	34	0,6125	44	0,01652	34	0,02499

Table 7. Co-authorship network employing centrality metrics

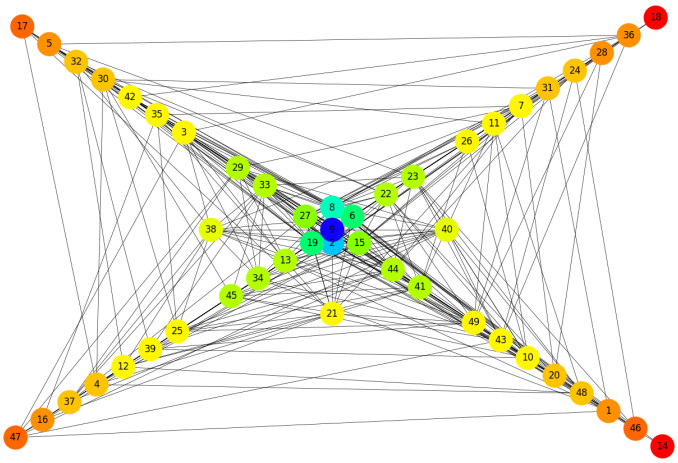


Fig. 7. Co-authorship network employing betweenness centrality metric

Table 8 shows the events in which the selected studies were published. Most of the studies were published in ACL and EMNLP conferences. Four out of the top five most cited papers were presented at ACL.

In summary, Klein and Manning's pivotal work [72] (S9) emerges as the most influential in UGI, cited by 43 works and holding an almost obligatory position in UGI research. Their earlier contribution [71] (S2) closely follows, earning citations from 27 works and pioneering performance improvements beyond right-branching. Headden et al.'s work [57] (S19) significantly impacts the field with citations in 26 works, advancing DMV through Bayesian inference and establishing it as the most influential Bayesian work in UGI. Kim et al.'s study [69] (S8) is acknowledged by nine works for introducing a neural parametrization applicable across neural network studies. Spitkovsky et al.'s work [125] (S15) is cited by 13 works, enhancing DMV methodologies through curriculum learning without additional complex techniques.

Table 8. Leading publication venues for grammar induction field

Event	Citations	Studies	μ citations
ACL	2112	15	132,0
EMNLP	480	12	40
NAACL	433	5	86,6
CONLL	218	4	54,5
EMNLP-CoNLL	30	1	30
Transactions of the Association for Computational Linguistics	75	3	25,0
AACL/IJCNLP	32	2	16,0
Journal of Machine Learning Research	121	1	121,0
Others	289	7	41.28

6.2 RQ2 - What approaches have been used in UGI?

The methods applied in UGI, both with constituent and dependency grammar, are divided into generative and discriminative categories. All discriminative models selected in this work use autoencoders where the encoder applies a discriminative approach and, in the decoder, a generative.

The generative approach is divided into three groups: structural, non-parametric, and parametric [141]. The structural generative approach, sometimes called *structural search* or *structural learning*, does not present a fixed grammar to be used by the model. It is learned, together with the model parameters, using heuristics. Early work on GI generally used the structural approach. [71]'s work contributed to the paradigm shift from the structural to the parametric model in UGI. S31 is the only selected work that uses the structural generative approach.

Parameterized models have a fixed number of parameters, such as μ (mean) and σ^2 (variance). However, it is not correct to say that non-parametric models do not have parameters, as it is not possible to determine the number of parameters of these models since the number of parameters grows as the amount of data also grows, and could be theoretically, infinite [45]. This approach allows adjustment of the model to its complexity and amount of data. The most used techniques in UGI tasks with a non-parameterized approach are the use of the Pitman-Yor [137] and Chinese restaurant [2, 102] processes.

Finally, the parameterized approach is the most widely used in UGI, accounting for over 90% of generative work. The parameterized approach starts with a fixed grammar model. The model parameters are then optimized to find the best syntax tree for a given sentence. The main techniques used in parameter optimization are Bayesian inference, the IO algorithm (which can be seen as an instance of EM) [8], and EM [34], and neural networks (including language models).

Most UGI models (94.1%) adopt a generative approach, with a minority (5.9%) opting for a discriminative approach. Within the generative paradigm, parameter search accounts for 93.7%, while non-parameter constitutes 4% and structural search constitutes 2%. The prevalent technique employed for grammar induction is Expectation-Maximization (EM), utilized in 56% of all studies. Despite recently using Neural Network and Pre-Trainend Language Models in grammar induction, it corresponds to 36%

Out of the total studies conducted, 34 (68% overall) center around dependency grammar. The Dependency Model with Valence (DMV) [72] is the most influential model for dependency grammar, which influenced 75.7% of dependency grammar studies. DMV is a generative parametric model EM algorithm to induce dependence. For constituency grammar, the Constituent Context Model (CCM) [71] is the most cited model applied to a constituent and also applies EM algorithm. There are other influential models cited in this study, such as U-DOP [17], CCL[113], DIORA [5], and ADIOS [123] (that was not selected in this mapping due rejection in Q3, but has

huge importance). In the following, we describe the studies in more detail. The studies are presented in Figure 8. We divided it into four groups: Expectation-maximization and IO, tree-substitution grammar (TSG) and DOP models, Bayesian inference, and neural networks. The colors discriminate the grammar representation used in the study in Figure 6: orange(Constituency grammar), pink (dependency grammar), and green (constituency and dependency grammar).

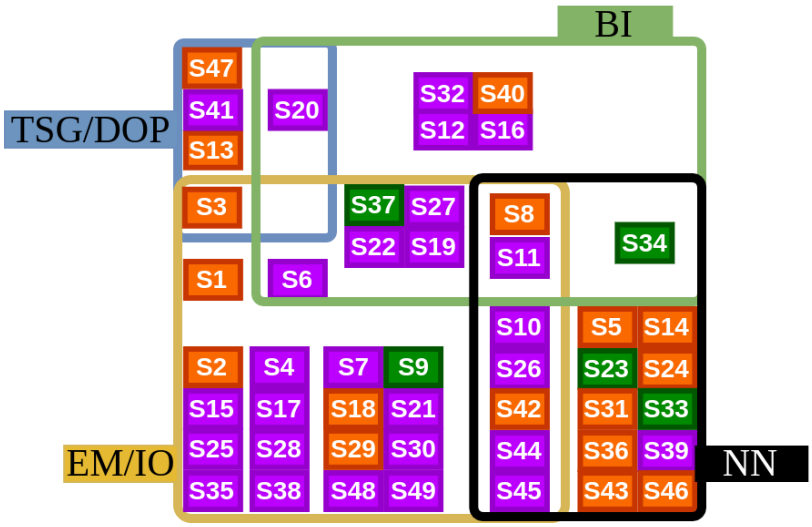


Fig. 8. Works organized by category implementation

6.3 Expectation-maximization

Inducing grammar presents a non-convex challenge. The EM algorithm offers an effective approach for approximating non-convex functions. This algorithm calculates maximum likelihood estimates from incomplete data, where the correct syntactic tree for a given sentence serves as the incomplete or unobservable data in the EM algorithm. Most studies employing the EM algorithm implement dependency grammar, with 72% adopting it, and 92% of those studies either implement or are based on DMV(S9) (S4, S6, S9, S10, S11, S15, S17, S19, S21, S22, S25, S26, S27, S30, S37, S38, S44, S45, S48, and S49). On the other hand, only 37% of the studies that apply constituency grammar use expectation maximization, and only four studies adopt CCM (S1, S2, S9, and S18). Several of these studies also utilized additional methods such as Bayesian inference (S19, S22, S27, S37) and neural networks (S11, S44, S45), among others (S31). Consequently, a comprehensive exposition of these studies will be provided in Sections 6.4, 6.5 and 6.6, respectively.

S2 (CCM) was the first unsupervised approach to surpass a right-branching baseline. This model leverages linguistic assumptions to induce grammar. The idea is to build the syntactic tree based on the context of the constituents. It uses spans to delimit the constituents in the sentence (bracketing) and EM to induce these bracketings, treating the sentence as observable and the bracketings as a latent variable. In contrast to the DMV model, only a few studies (S1, S2, S9, and S18) are based on or utilize the CCM. All employ the classical expectation-maximization approach. CCM encounters a challenge from the high sparsity of data due to the exponential increase in spaces between words during tree construction. To mitigate this growth, S1 employs a log-linear model. In contrast to the Expectation-Maximization (EM) algorithm, S1 adopts an alternative estimation

method, as detailed in [81]. Unlike other studies, S18 focuses on grammatical classes as markers for the ends of sentences, such as modal verbs (*could*, *can*, *should*, e.g.). That study defines a total of 22 POS tags that can serve as separators and eight POS tags as sub-separators. The study applies only to the English language, which makes it challenging to analyze the effectiveness of this method to different languages.

As previously mentioned, DMV (S9) is the most influential work in grammar induction. In contrast to CCM, DMV adopts a top-down approach, generating the parsing tree recursively. S9 uses valence (the number and type of argument controlled by the predicate; among the most common are transitive and intransitive verbs [3]). The idea behind the DMV model is to control the generation of the tree, which, for each branch to be generated, uses probability distributions to make decisions on when to generate ($P_{STOP}(\neg STOP|h, dir, adj)$) and which branch to generate ($P_{CHOOSE}(a|h, dir)$). The variables *h*, *dir*, *a* and *adj* are respectively the head of the relation, the direction in which the argument will be generated - right or left, the argument to be generated and whether the argument has already been generated in the tree in the *dir* direction. Ten studies (S9, S10, S11, S15, S22, S26, S38, S44, S45, S48) studies based on DMV use classical EM, but only four studies (S9, S15, S38, S48) do not additionally other approaches, such as Neural Networks and Bayesian Inference. S15 implements EM in conjunction with curriculum learning [9], a method that initially trains the model on more straightforward data and gradually introduces more complex data until the entire database is covered to improve performance dependency grammar induction. Differently from DMV, S38, the POS tags induction from clustered words allows words to have different POS tags in other contexts. Additionally, this work employs punctuation strategies to enhance performance, a technique also employed by S6, S7, S25, S29, and S35. S48 uses a stack automaton-based parser to identify center *embedding* [25] in different languages. Embedding refers to a feature of natural language that allows the insertion of clauses within sentences without grammatical loss. When that clause insertion occurs in the center of the main clause, it is called *center-embedding*. Identifying this syntactic construction is quite complicated. S48 study showed that performance in different languages varies considerably. S29 and S35 are studies that apply the classical EM algorithm but do not model DMV or CCM. S29 Uses the finite state model in a probabilistic way. Finite state models, also known as *chunking* (local structures), are an alternative to using CFG [58]. This allows the induction to be done partially and then combined using cascading (a strategy that allows *chunks* replacements with other words forming "multi constituents" contributing to constructing the syntactic tree). To perform partial induction, S29 uses the HMM and PRLG (right recursion grammar that uses the HMM). See [122] and [44] for further discussion. S35 proposed three models for UGI based on guesses of distinguishing the beginning and end of a sentence. The first model is based on the distribution of word adjacents in the sentence. The second model concerns word classes. Finally, the third model uses punctuation.

Training long sentences poses a challenge to EM, primarily due to the prominent issue of local maxima [34] [131]. That problem contributed to the proposition of some EM variations. Several studies have implemented these modified approaches:

- Viterbi-EM, introduced by Spitkovsky et al [131], known for its expedited and accurate estimation capabilities, has demonstrated its effectiveness in studies S25, S28, S38, S44, S45, and S49 for training extended sentences.
- Lateen-EM, proposed by Spitkovsky et al. [126], strategically employs a deliberately slower convergence rate to circumvent local maxima, resulting in enhanced training outcomes as observed in studies S6, S7, and S21.
- PR-EM, introduced by Ganchev et al. [48], incorporates constraints on the posterior distributions of latent variables within the EM framework, yielding notable improvements in S30.
- Contrastive estimation, proposed by Smith and Eisner [119] applied in studies S4 and S17, is a viable means of augmenting the efficacy of sentence training procedures.

The Virtebi-EM, as introduced by Spitkovsky et al. [131] (S49), employs the Viterbi algorithm—a dynamic programming technique commonly associated with HMM for computing posterior probabilities [41]. This

approach serves as a replacement for the IO Algorithm. This replacement offers more speed and simplicity and outperforms the classical EM. This approach is referred to as *"Hard EM"* or *"Viterbi EM"*. See Samdani et al. [110] for further discussion. Among the works reviewed, S25 was the primary work leveraging punctuation as boundary identification. That study suggests that punctuation improves UGI in dependence grammar. Notably, S28, the sole study selected in our systematic mapping, opts for CCG [132], a lexicalized grammar formalism, instead of CFG. This framework is relatively uncommon in UGI research. This study demonstrates that Viterbi EM performs less than Classical EM with CCG. Furthermore, the study indicates that CCG can yield results comparable to those of using CFG in grammatical induction.

Lateen-EM proposed by Spitkovsky et al. [126] (S21), contrastive estimation (a generalization of EM) [119] and posterior regularization are less frequent variations of EM. Lateen-EM alternates between *"hard"* and *"soft"* EM. It is connected to a significant variation of the EM algorithm known as softmax [140]. Although this systematic mapping did not uncover that specific study, its citation holds importance. Softmax, differently from Lateen-EM, keeps fixed between *"hard"* and *"soft"*. S6 introduces a complex network structure (non-neural), enabling diverse combinations and transformations of different distributions for inducing grammar. That network is built upon the DBM models proposed by S35. S7 implements S35, adding capitalization to identify boundaries. That study reveals significant disparities between languages, even within different textual genres of the same language. Contrastive Estimation is employed in S4 to penalize long dependencies. In this study, the distance is calculated as the exponential sum of differences between the lengths of the sentence's dependencies (the interval of words between the dependent word and the head of the relation) to define a long dependency. S17 applies Contrastive Estimation to compute log-linear models, as using EM for these models is impractical. The study demonstrates that CE proves valuable in guiding the learning process. S30 employs a sparsity bias in dependency grammar using posterior regularization.

6.4 Neural Networks

Inducing grammar in long sentences without supervision poses a significant challenge, particularly in dependency grammar where neural networks must effectively retain information across extended distances [63][109]. As the dependency distances grow longer, the complexity of the sentence increases [82]. Before the advent of the seq2seq model [136], most papers limited their analysis to no more than ten tokens for each sentence. In our investigation, we reviewed eighteen studies employing neural networks, all published post-2013. Notably, a substantial majority (89%) emerged after introducing the transformers model [145].

According to our mapping, Yang et al. [150] (S45) was the first study to apply neural networks to UGI efficiently. Although S39 is a study published before S45, it uses a neural network to supervise grammar induction [78]. We categorized the works on neural networks obtained in this study into five groups: those applied to PCFG (S8, S23, S24, S34, S43), those related to language models (S5, S14, S26, S33, S34, S36, S46), those adapted from DMV (S10, S11, S26, S44, S45), and those related to chart models (S8, S14, S42). Additionally, there exists another work referenced as S39, which may not fall into these designated groups.

S8 and S43, in our mapping review, were the primary works to apply neural networks with PCFG for inducing unsupervised grammar. S8 employed a compound probability approach (where the parameters are treated as random variables [108]) to limit the number of rules generated. They introduced a neural parametrization that relies on distributional representations of PCFG rules, which is particularly beneficial for large grammars. Due to the lack of context in the constituency grammar syntactics, S43 used ELMO to make the constituency grammar IGNS more semantic. The *embeddings* of each ELMO word are used as observable variables. In this study, to generalize to rare words, they use *normalizing flow*(a machine learning technique to transform a simple distribution into a complex distribution [107]). In neural Networks based on normalizing flow, the mapping from input to output is bijective; that is, the number of output neurons must be equal to the number of input, but unlike

autoencoders, the output and input are different because the goal of this network generates a more complex distribution from a simple one. This type of network allows computing the posterior in an implicit way where the computation is given in both directions, which, unlike neural networks that only employ backpropagation [6] [107]. S34 introduced a lexicalized neural model employing the Monte Carlo method [85]. This study extended the work of S8 and implemented S9 strategies, integrating models for constituent and dependency grammar. That allows exploring the benefits of unifying constituent and dependency grammar in a single model, enabling for the simultaneous induction of both grammars. To gather distributional information, S34 utilized Glove embeddings [100]. S23 further expanded on the research of S34 by incorporating lexical grammar, which defines grammatical rules involving multiple words with reciprocal influences among the words. Refer to Jason Eisner [37] for a detailed discussion. This type of grammar incurs a high computational cost, with a complexity of $O(N^5)$ for parsing using the CKY algorithm and $O(N^4)$ in specific improvements [38]. To mitigate computational expenses, S23 implements tensor decomposition [73]. S24 shows that using a more significant number of symbols (nonterminal and preterminal) improves performance in constituency grammar. They achieve this improvement by employing a new neural parametrization based on tensor decomposition.

S10 and S44 pioneered word embedding as a representation in grammar induction models. Nevertheless, these models fall behind in capturing nuanced meanings in various contexts. According to our mapping review, Kim et al.'s study (S5) [68] stands out as the primary exploration into language models for grammar induction. This study introduced a technique for extracting constituent trees from language models, incorporating syntactic distance [114]. S14, S33, and S36 also employ this technique; moreover, all three contain attention models and masking language models. S33 is the only study retrieved in this work that uses large language models applied to dependency grammar. It uses a similar approach to S34 by simultaneously inducing dependency and constituency grammars. To model dependency grammar with language models, S33 utilizes syntactic height [86]. S36 introduces two models: Unsupervised Parsing Outside Association (UPOA) and Unsupervised Parsing Inside Association (UPIO). The study also incorporates few-shot learning (FPIO and FPOA), though, in this work, we do not discuss these few-shot models due to their semi-supervised nature. S36 computes span scores from the attention matrix rather than hidden representations used in earlier models. S14 leverages R2D2, a bidirectional language model employing LSTM [60]. The core idea of this work is to use this model for fine-tuning (on raw unannotated text) based on tree extraction from the CKY table. S26 adopts an interesting strategy, incorporating first-order nodes of the tree (child, parent, sibling) and second-order nodes (grandparent, grandchild, nephew, cousin) to contribute to grammatical induction. This study leverages the neural networks proposed by S45 and S11. S46 implements the concept of constituent testing, a method rooted in generative grammar, particularly in transformational grammar [21]. This study incorporates strategies from S2, such as the utilization of sentence space, as well as the exclusive use of binary trees. The primary approach of the model is to leverage RoBERTa [83] for the real/false task. This task combines constituent testing to predict whether a given syntactic tree corresponds to a provided sentence. Although language models have produced fascinating results, investigations are confined to languages with accessible large-scale trained models. Conventional methods like Bayesian inference and various forms of Expectation Maximization continue to hold significance in research.

As mentioned before, S45 was the first model to implement neural to unsupervised grammar induction tasks. This study incorporates DMV into a neural network (NDMV): a relatively straightforward network comprising only three layers (input, hidden, and output). The input layer receives information on the valence and POS tags of the word. The hidden layer determines the direction in which a branch will be generated and whether it will indeed be generated. S10 builds upon S45 by incorporating a lexicalized neural network (L-NDMV) that provides richer information. Non-terminals in lexicalized models using CFG also incorporate the word associated with its parent (head in dependency grammar). S11 is a discriminative as well as a generative model that extends NDMV. This study implements autoencoders to use generative and discriminative approaches. Where the encoder is discriminative, and the decoder is generative. Unlike S43, applied normal flow with PCFG, S44 was applied to

DVM. S44 used this technique with skip-gram. They showed that using this type of network is more efficient for computing the prior.

S42 employs the Inside-Outside algorithm to induce grammar by populating charts in CKY parsing (Kasami, 1966; Younger, 1967). The resulting syntactic tree is constructed from subtrees. For model training, they utilize an autoencoder, which predicts each word in the syntactic tree using a masked context. Although we came across a more advanced version, S-Diora, in our mapping, it was ultimately excluded due to its narrower focus on parsing [35]. S39 is one of the early endeavors to apply a supervised approach to unsupervised induced grammar. They employ the concept of integrated reranking; training supervised parsers with unannotated data. In this model, the parser induces grammar to generate multiple syntactic tree candidates, and the integrated reranking process selects the best one. In contrast to S42, which employs an autoencoder, S39 utilizes an Inside-Outside recurrent neural network (IORNN) for model training. Notably, the IORNN constructs the tree in a top-down fashion.

6.5 Bayesian Inference

Since unsupervised grammar induction treats parse trees as unobservable variables, just like EM, Bayesian inference is an excellent strategy to infer the distribution of this variable, especially when this variable can be treated as posterior or priori. The reference of these distributions can be calculated through approximation. There are two main strategies: sampling and Variational Inference (VI) [13]. Five studies used sampling: S12, S16, S32, S34, and S40. Variational inference is applied in S8, S11, S19, S22, S27, S34 and S37.

S16 introduces constraints in sampling to improve performance in Czech. These constraints concern root structure and initialization to remove cycles. They experimented with different settings to choose the best one for Czech and generalized the setting to another 18 languages. S12 extends S16, but it guarantees the generation of projective trees, unlike S16. S12 uses the hypothesis that words can be removed without altering the syntactic correctness of the sentence, known as the *reducibility principle* [84]. The more reducible a token is, the greater the chance it is a dependent term. Based on DMV ideas, S12 introduces the fertility model to replace STOP distribution in the DMV model; the idea is to generate children from both the right and left and then find their dependents. They also applied a subtree model that analyzes the tree in multiple order (parents, sons, e.g.). To find the dependents, they use Gibbs sampling. S32 extends S12 to apply stop distribution to very large corpora. S40 uses Gibbs sampling and cognitively motivated bounds on recursion to limit the search to induce PCFG. Differently from the classical models that use top-down or bottom-up parsing, S40 uses a left-corner parser that is able to process in some order. See [106] and [143] for further discussion.

S19, known as *Extended Valence Grammar – EVG*, is the most well-known and influential study that used DMV ideas. EVG, unlike DMV, uses different distributions to represent valence and Variational inference with CFG and Dirichlet. Despite being a very relevant study and one of the most cited, only S22, S44, S37 and S25 use this method. Variational inference is also used by S22, who, instead of using Dirichlet to estimate the priori, opted for Logistic Normal – LN. S22 found that using LN, it is possible to gain performance compared to using the Dirichlet distribution. However, the variation is smaller when considering sentences of all sizes. S27 applies a generalization of S22 by using a distribution family of priors (a distribution over a collection of multinomials). Initially, there were reservations about including S37 in this mapping due to its utilization of the duration time of spoken words as an additional input characteristic for the model. However, considering that sound contributes to phonology, an integral aspect of grammar, and this data undergoes no prior training, we decided to retain this study. The sound data used in the study was sourced from the CHILDES database [87], a spoken corpus of children, and the model employed aligns with the one proposed by S19.

6.6 Other approaches

Some works use different approaches. Among them are works related to the DOP model (S3, S20, and S47), Tree Substitution Grammar (S20, and S41), Additive Metrics (S31) and CCL (S13).

The DOP model seeks to construct trees by combining subtrees. The model uses corpus statistics from an annotated treebank to build trees. S3 uses word substitution, but in binary trees, and then computes the probability of this tree. To estimate the best tree, since it is an NP-complete problem, the 100 most probable trees are considered, which are chosen using the algorithm *viterbi* [41]. In addition to this model, another model is presented, the ML-DOP, which uses EM. Although S47 presents practically the same method used by S3, S47 seeks to induce grammar in a generalized way without considering whether the grammar is constituent or dependent.

S41 and S20 are the only non-parameterized studies selected in this systematic mapping. Both use DMV and DOP [15] applied to the *Pitman-Yor* [103] algorithm (an algorithm that uses the *Poisson-Dirichelet* distribution), which produces power distributions closer to those existing in language Natural.

S31 is the only study we selected that uses the structured generative approach; they do not use a grammar model. S31 model CCM uses bilexicate grammar, which, instead of using tensor decomposition for parameter estimation like S23, S31 uses spectral model [4] based on additive metrics.

S13 was the first work to induce grammar entirely using raw text without using gold POS tags or induction. S13 introduces the common cover link, a different representation of constituency grammar. Training is done incrementally, generating links even before the sentence is completely read. The work takes advantage of the Zipf law to induce grammar. S29 extended the CCL with more robust techniques.

Tables 9 and 10 display the performances of all studies included in this mapping review that employ constituency and dependency grammar, respectively. For constituency, grammar was the F1 measure, whereas, for dependency, grammar was the DDA measure. The first column showcases scores for words tested on WSJ10, while the second column presents results on WSJ∞. The legend W/GP denotes works utilizing gold POS tags, whereas W/O signifies those not using gold POS tags.

6.7 RQ3 What resources and methodology are applied in the studies?

The selected studies have a great diversity of resources and methodologies. This section will be divided into the following Sections: data, training, and testing constraints. In data, we will focus on issues such as the primary corpus and treebanks used and the size of these data. In training restrictions, we will deal with data pre-processing, using lowercase, punctuation, use or not of gold POS tags, the number of words per sentence in training, and the number of words per test sentence.

6.8 Data

During the selection phase, we identified various sources for corpora and treebanks utilized by researchers. These included corpora generated through sampling from other corpora, synthetic corpora, and corpora constructed by the authors of the studies. Distinctive corpora like CHILDES [87] and ATIS [142] were present within the chosen studies. Researchers who employed these particular corpora may have also tested their models with additional corpora for inclusion in this mapping. Among the most commonly used treebanks, we have the Penn Treebank (PTB), known as WSJ in UGI studies because it consists of a subset of PTB featuring texts from the Wall Street Journal (WSJ) [88]. Additionally, researchers made use of the Chinese Treebank (CTB) [147], the NEGRA Treebank [117], and Universal Dependencies (UD), including versions from CONLL 2006 and CONLL 2007 [19]. Although the initial three corpora were designed as Treebanks of constituents, they are commonly used to induce dependency grammar. This is accomplished by converting these corpora into dependency grammar through a parsing converter, as demonstrated in the work of Collins et al. (1999) [30] and [148] algorithm.

Table 9. Performance in different studies on constituency grammar on WSJ treebank using F1 measure

	WSJ10		WSJ∞	
Study	W/GP	W/O	W/GP	W/O
S2	70,9	63,2		
S3	82,9		66,4	
S5				48,3
S13		75,9		57,4
S14				57,3
S18	74,6			
S23				60,4
S29		72,1		54,2
S31	69,2		45,5	
S33				54
S34				55,3
S36		56,9		
S40		73,4		54,1
S42		68,5		60,9
S43		56		38,6
S46				62,8
S47	78,5			

Figure 9 shows the most used corpora. They follow the pattern of the previous Figures 6 and 8. It can be observed that despite the WSJ being a Treebank of constituency grammar, most of the selected studies are applied to dependency grammar. Since WSJ has become a Benchmark in the UGI task, most studies use it to compare with previous studies. The corpus presented in Figure 9, only CONLLX (CONLL 2006 and CONLL 2007) and UD are intended for dependency grammar.

The first treebank designed for dependency grammar, known as the *Prague treebank*, was constructed for the Czech language [**?**]. However, despite being the pioneering treebank for this grammar framework, it does not hold a prominent position among the most widely utilized corpora. One contributing factor to its comparatively lower usage, in contrast to corpora such as CTB or NEGRA, could be attributed to the extensive inclusion of over 3000 POS tags in the *Prague treebank*, unlike UD, which adopts a more concise set of 37 tags.

The WSJ comprises 23 sections, each categorized based on the size of the sentences it encompasses. Authors adopt the standard nomenclature WSJX, with X denoting the maximum length of sentences in a given section. For instance, WSJ10 signifies that all sentences in that section contain, at most, ten tokens. Typically, training on WSJ involves utilizing sections 2 (WSJ2) through 22 (WSJ100). Section 23 is specifically designed for testing sentences of varying lengths, with authors often imposing restrictions on the sentences under examination. When all sentences in section 23 are employed without length constraints, the terminology WSJ∞ is commonly used.

The CTB was the first annotated corpus for constituency grammar that is unrelated to the Greek or Cyrillic written systems. This corpus contains 1, the size of the WSJ, and an average of 28.7 token pair sentences. As the WSJ, CTB contains news from political and economic sources.

CONLLX is a compiled corpus of 19 languages, 13 in the 2006 version (German, Arabic, Bulgarian, Chinese, Danish, Slovenian, Spanish, Dutch, Japanese, Portuguese, Swedish, Czech, and Turkish) and 6 in the 2007 version do not exist in the 2006 version (Basque, Catalan, Greek, Hungarian, English, and Italian). Some of these corpora were integrated into the UD.

Table 10. Performance in different studies on dependency grammar on WSJ treebank using DDA measure

	WSJ10		WSJ∞	
Study	W/GP	W/O	W/GP	W/O
S9	47,5	42,3		
S30	64,4			
S38				59,1
S17	49			
S20	66,4		53,4	
S37	52,5		68	
S34		40,5		
S4	56,6			
S6	72,2		64,4	
S10	75,1		59,5	
S11	75,6		61,4	
S15	57,1		45	
S19	68,8			
S21			55,6	
S22		59,4		40,5
S25		69,5		58,2
S26	79,9		67,5	
S27	62		42,2	
S39	72,7		66,2	
S44		60,2		47,9
S45	72,5		57,6	42,7
S49	65,3		47,9	

Despite the existence of corpora, including multiple languages, most studies focus only on English, accounting for approximately 42% of all investigations. Among the studies covered in this mapping review, English is used in all studies, followed by German at 44.8%, Chinese at 32.6%, and Portuguese and Swedish at 26.5%. Despite over 30 languages being used in different studies, only 15 are employed for constituency grammar, with Portuguese exclusively utilized for dependency grammar. Approximately 42% of the studies concentrate on a single language, 53% on at most two languages, and merely 22% are trained across more than ten languages. Furthermore, only 20% of the studies apply constituency grammar to more than eight languages. Table 11 displays the languages utilized in the studies. The first and fifth columns indicate the languages employed, while the second and sixth columns represent the number of studies utilizing constituency grammar. The third and seventh columns depict the studies employing dependency grammar, and the fourth and eighth columns show the total number of studies for each language.

6.9 Raw text X Gold POS tags

Most studies use gold POS tags to train their models. Some use only the POS tag with input, others use both the token and the POS tags, and some studies use extra information, such as acoustic cues. Studies that use gold POS tags show better results than those that use raw text. Despite not directly using gold POS tags, the language models applied to grammatical induction benefit from a gigantic corpus that indirectly induces POS tags through distance and syntactic height [86, 114].

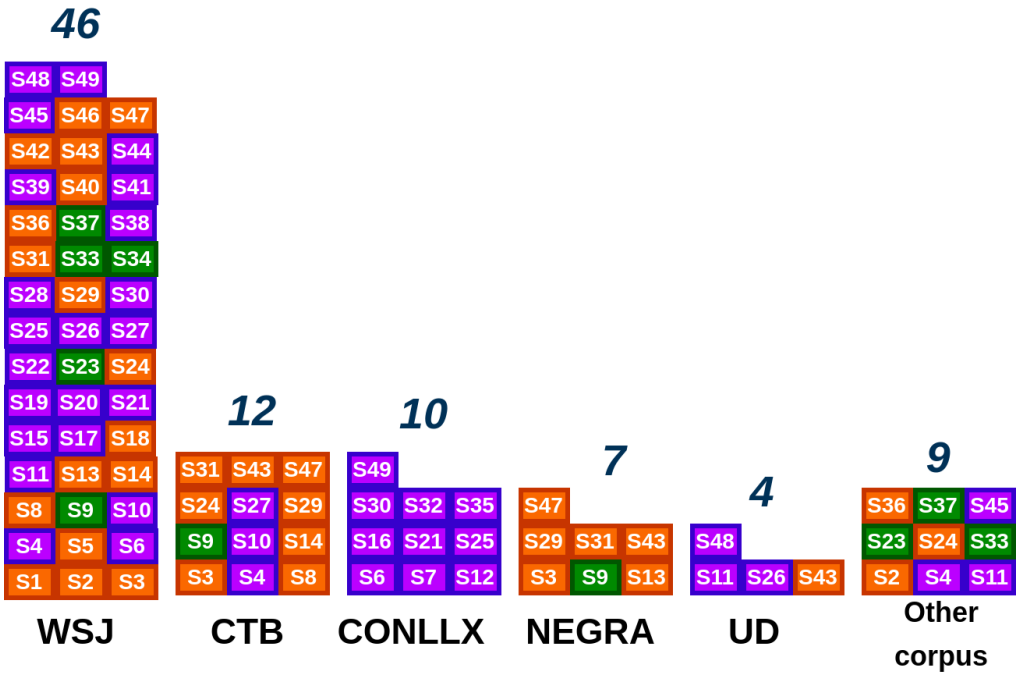


Fig. 9. Corpora most common in the studies

6.10 Training and testing constraints

In training restrictions, a common practice in most studies is the limitation of sentences with up to 10 tokens in the testing phase. This occurs because the longer the sentence, the greater its complexity, especially for dependency grammar. By calculating the average of the results of the 49 selected studies (without considering the language that was trained or whether the POS tags were noted) using the metrics F1 (constituency grammar), DDA (dependency grammar considering the tree generation direction), and UDA (dependency grammar without considering the generation direction), we obtained the following values which are presented in Table 12

Other restrictions include removing punctuation (performed by 93.61

6.11 RQ4 Does language influence different approaches?

S6 is an exciting work to analyze this research question. This study tested five different models in 19 languages. The variation in performance between languages is almost twice, considering other systems. However, some languages, such as Arabic, Japanese, and Bulgarian, present more sensibility to different models. On the other hand, some languages are more regular, such as Chinese, Swedish, and Czech. In studies that use corpora from several languages, the standard deviation can vary from 10 to 16 points, depending on the metric used and the length of the sentences. These data only present suggestions, but they are not enough to conclude since the amount of data is small to carry out a robust and reliable meta-analysis.

A more reliable analysis is only possible from the analysis of experiments for Chinese and English, as presented in Table 13. The data presented in the table were compiled from all experiments in the 49 studies. The metrics'

Table 11. Use of different languages by grammar framework

Language	C	D	S	Language	C	D	S
English	22	32	49	Catalan	0	8	8
German	9	14	22	Japanese	0	7	7
Chinese	9	8	16	French	2	3	5
Portuguese	0	13	13	Polish	2	2	4
Swedish	2	11	13	Hebrew	2	1	3
Basque	2	10	12	Korean	3	0	3
Czech	1	11	12	Finnish	1	1	2
Danish	0	12	12	Slovak	0	1	1
Dutch	0	12	12	Russian	1	0	1
Bulgarian	0	11	11	Uyghur	1	0	1
Spanish	0	11	11	Croatian	0	1	1
Turkish	0	10	10	Estonian	0	1	1
Hungarian	2	8	10	Hindi	0	1	1
Italian	0	10	10	Indonesian	0	1	1
Slovenian	0	10	10	Latin	0	1	1
Arabic	0	9	9	Norwegian	0	1	1
Greek	0	9	9	Persian	0	1	1

Table 12. Average score in all languages

Grammar	**Metric**	**<= 10 tokens**	∞
Constituency	F1	59.31	40.67
Dependency	DDA	51.90	39.68
Dependency	UDA	62.48	52.70

mean (μ) and standard deviation (σ) were considered. For dependence, DDA and UDA were computed and considered together to calculate the mean and standard deviation. Fields with - had less than three experiments performed in the 49 selected studies.

Constituency grammar yields superior results in English, while dependency grammar performs optimally in Japanese. Notable disparities are apparent among languages regarding dependency grammar, whereas such variations are less pronounced in constituency grammar. These findings imply that certain languages may be more compatible with specific models. Nonetheless, it is important to note that there is insufficient data to confirm this hypothesis definitively.

6.12 RQ5 - What evaluation metrics are used?

In assessments focusing on constituent grammar, the F1 metric is commonly employed. While some studies utilize PARSEVAL as a distinct metric [105], it is essentially an F1 measure. Evaluating dependency grammar involves two metrics: one measures the accuracy of relations without considering their direction (head, dependency), and the other considers the direction of the relation. Interestingly, different studies may use varied terms for the same type of assessment, such as head attachment accuracy, directed accuracy, directed attachment accuracy, directed dependency accuracy, and unlabeled directed attachment. Despite the nomenclature differences, they all pertain to accuracy considering the direction of the relation. A directed accuracy score is consistently applied across all

Table 13. Performance in different languages

	Constituency				Dependency			
	10 tokens		∞		10 tokens		∞	
Língua	μ	σ	μ	σ	μ	σ	μ	σ
Arabic	-	-	42,65	6,17	-	-	21,65	7,59
Basque	-	-	42,85	7,9	-	-	34,08	6,71
Bulgarian	-	-	48,93	12,97	-	-	47,5	9,51
Catalan	-	-	-	-	-	-	55,9	8,16
Czech	-	-	46,26	10,89	-	-	40,7	8,83
Chinese	48,85	6,7	49,02	10,83	34,8	4,17	47,43	11,88
Danish	-	-	44,01	8,5	-	-	35,35	9,09
Dutch	-	-	43,89	6,16	-	-	38,2	11,48
English	**66,94**	8,71	**59,05**	10,85	**52,02**	5,97	48,24	9,23
Eslovenian	-	-	45,74	9,52	-	-	34,42	11,42
German	60,92	5,51	51,89	10,85	-	-	39,92	11,66
Greek	-	-	-	-	-	-	25,35	11,62
Hugarian	-	-	-	-	-	-	34,9	20,35
Italian	-	-	-	-	-	-	37,6	4,15
Japanese	-	-	58,4	7,72	-	-	**55,02**	5,3
Portuguese	-	-	48,77	15,89	-	-	46,2	17,04
Swedish	-	-	49,64	9,09	-	-	44,72	4,74
Spanish	-	-	58,85	11,21	-	-	41,25	16,62
Turkish	-	-	41,04	12,63	-	-	24,83	8,59

studies dedicated to dependency grammar. In this study, we adopt DDA (Directed Dependency Accuracy) for assessing directed accuracy and UDA (Undirected Dependency Accuracy) for evaluating undirected accuracy.

6.13 RQ6 - What are the trends in the field?

Over the past two decades, the WSJ treebank has been a foundational English reference point. Its continued relevance in English language studies persists due to the extensive utilization of this dataset in numerous research, including those focused on dependency grammar. Introducing the Shared Task CONLLU 2006/07 [19] was crucial in diminishing the English-centric dominance within the UGI. The graphical representation in Figure 10 illustrates the dynamic evolution of multiple languages within the UGI. This observation highlights a discernible inclination toward incorporating a diverse array of languages (more than 5) in linguistic studies. Notably, approximately half of the studies in dependency grammar in this work employ more than five languages; in the last decade, it constituted 61%. In contrast, only around 16% of studies in constituency grammar explore more than five languages, and all of these studies were published within the past five years.

During this mapping review, we have found many works on cross-linguistic and transfer learning. Some results on transfer learning suggest this initiative as an alternative to mitigate the lack of data [49]. The majority of these studies were published after 2018. This method could be a trend for some languages that increase performance.

A prevailing trend involves the utilization of extensive language models, particularly in the context of constituency grammar, for the extraction of syntactic structures. However, this approach is not without its limitations. The primary constraint is the requirement for a substantial volume of data, posing challenges for languages with limited available data. Furthermore, the outcomes lag behind traditional methods that eschew large language

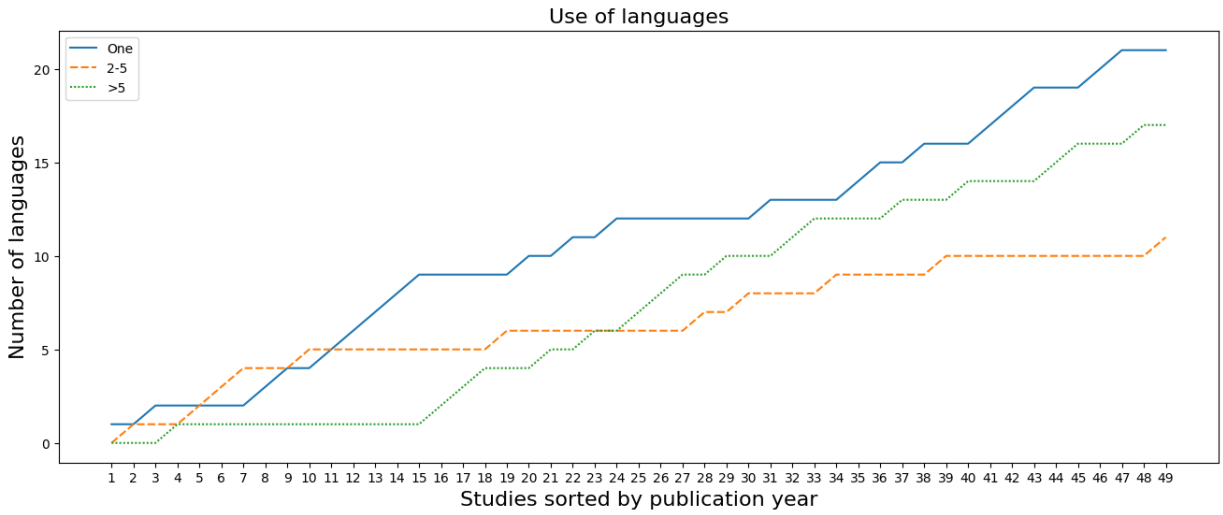


Fig. 10. Use of multiple languages across the years.

models [68]. Lastly, despite advancements in neural networks and expansive language models, there persists a challenge in comprehending the intricacies of dependency grammar within the realm of grammar induction.

The use of raw text to induce grammar is a trend for the following years due to the relevance of languages with few resources.

7 DISCUSSION

UGI is a highly complex task. That implicates a proposition to a wide range of various models, techniques, and strategies to UGI. Based on the selected studies, we have not identified an academic consensus on which models to use. Many studies use unique methods to solve specific problems but do not allow generalization to other languages. Although grammatical induction has its roots in language acquisition, there are few attempts to include ideas from linguistics and psychology in UGI models. Most works look to machine learning for techniques that can improve UGI task performance. For constituency grammar, S3 still has today the best results for sentences at most ten tokens. However, it can not be scaled for longer distances due to the increase in complexity.

When looking less closely, the reader is mistakenly led to see that the area is using “brute force” by trying many different models and strategies to solve the problem. However, that sea of models and techniques could be grouped, most of them, into only four groups: studies that apply EM, Bayesian Inference, Neural Networks, and methods that use heuristics such as DOP and TSG. Variations occur within these groups. Despite EM being the first algorithm used in UGI, it is still used alongside new methods, such as Neural networks. Despite advances in neural networks in various applications such as translation systems, computer vision, speech processing, and text generation, they present no significant differences in UGI concerning traditional models.

Recently, large language models have been applied in UGI. It gives an exciting insight into the field. However, the use of large language models does not present good results compared to other subfields in NLP, such as text generation and machine translation. Furthermore, it is applied almost exclusively to constituency grammar S33 is the only work that uses large language models to induce dependency grammar. However, it depends upon constituency grammar to generate dependency distribution.

We noted many different corpora used in grammar induction. Among them, WSJ has become the benchmark to compare to previous works. Some initiatives, such as UD [32], still have difficulty being inserted.

The systematic mapping did not recover some critical studies; we must cite them here. Zannen [144] applied the alignment technique used in ancient translation systems to induce grammar in an unsupervised way. One of the reference works in UGI that uses clustering was proposed by Clark et al. [26]. The proposal was based on the distributional hypothesis, in which words with similar contexts tend to occur together [55]. [26], showed high mutual information between a constituent and the previous and subsequent tokens. Finally, Shen et al. [114] proposed a new neural language called *"Parsing-Reading-Predict Networks"* (PRPN). [114] introduce the concept of syntactic distance. This concept is used in most works that apply large language models to extract structures.

8 CONCLUSION

ACKNOWLEDGMENTS

-

REFERENCES

[1] Adrian Akmajian, Ann K Farmer, Lee Bickmore, Richard A Demers, and Robert M Harnish. 2017. *Linguistics: An introduction to language and communication.* MIT press.

[2] David J Aldous, Illdar A Ibragimov, Jean Jacod, and David J Aldous. 1985. *Exchangeability and related topics.* Springer.

[3] David J Allerton. 1982. Valency and the English verb. *(No Title)* (1982).

[4] Animashree Anandkumar, Daniel J. Hsu, and Sham M. Kakade. 2012. A Method of Moments for Mixture Models and Hidden Markov Models. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland (JMLR Proceedings, Vol. 23)*, Shie Mannor, Nathan Srebro, and Robert C. Williamson (Eds.). JMLR.org, 33.1–33.34. http://proceedings.mlr.press/v23/anandkumar12/anandkumar12.pdf

[5] et al. Andrew. 2019. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. *NAACL-HLT* 38 (2019), 453–468.

[6] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. 2019. Analyzing Inverse Problems with Invertible Neural Networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rJed6j0cKX

[7] Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, 3011–3020. https://doi.org/10.18653/v1/2021.eacl-main.262

[8] James K Baker. 1979. Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America* 65, S1 (1979), S132–S132.

[9] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

[10] Robert C Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the stimulus revisited. *Cognitive Science* 35, 7 (2011), 1207–1242.

[11] Alan W Biermann and Jerome A Feldman. 1972. A survey of results in grammatical inference. In *Frontiers of pattern recognition*. Elsevier, 31–54.

[12] Yonatan Bisk and Julia Hockenmaier. 2012. Simple robust grammar induction with combinatory categorial grammars. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 26. 1643–1649.

[13] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association* 112, 518 (2017), 859–877.

[14] Phil Blunsom and Trevor Cohn. 2010. Unsupervised Induction of Tree Substitution Grammars for Dependency Parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1204–1213. https://aclanthology.org/D10-1117/

[15] Rens Bod. 1992. A Computational Model Of Language Performance: Data Oriented Parsing. In *14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, August 23-28, 1992*. 855–859. https://aclanthology.org/C92-3126/

[16] Rens Bod. 2006. An all-subtrees approach to unsupervised parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. 865–872.

[17] Rens Bod. 2006. Unsupervised Parsing with U-DOP. In *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL 2006, New York City, USA, June 8-9, 2006*, Lluís Màrquez and Dan Klein (Eds.). ACL, 85–92. https://aclanthology.org/W06-2912/

[18] Joan Bresnan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler. 2015. *Lexical-functional syntax*. John Wiley & Sons.

[19] Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL 2006, New York City, USA, June 8-9, 2006*, Lluís Màrquez and Dan Klein (Eds.). ACL, 149–164. https://aclanthology.org/W06-2920/

[20] Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised Parsing via Constituency Tests. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4798–4808. https://doi.org/10.18653/v1/2020.emnlp-main.389

[21] Andrew Carnie. 2021. *Syntax: A generative introduction*. John Wiley & Sons.

[22] Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on information theory* 2, 3 (1956), 113–124.

[23] Noam Chomsky. 1957. Syntactic Structures. (1957).

[24] Noam Chomsky. 2002. *On nature and language*. Cambridge University Press.

[25] Noam Chomsky. 2014. *Aspects of the Theory of Syntax*. Vol. 11. MIT press.

[26] Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning, CoNLL 2001, Toulouse, France, July 6-7, 2001*, Walter Daelemans and Rémi Zajac (Eds.). ACL. https://aclanthology.org/W01-0713/

[27] Shay B. Cohen, Kevin Gimpel, and Noah A. Smith. 2008. Logistic Normal Priors for Unsupervised Probabilistic Grammar Induction. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou (Eds.). Curran Associates, Inc., 321–328. https://proceedings.neurips.cc/paper/2008/hash/f11bec1411101c743f64df596773d0b2-Abstract.html

[28] Shay B. Cohen and Noah A. Smith. 2009. Shared Logistic Normal Distributions for Soft Parameter Tying in Unsupervised Grammar Induction. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*. The Association for Computational Linguistics, 74–82. https://aclanthology.org/N09-1009/

[29] Trevor Cohn, Phil Blunsom, and Sharon Goldwater. 2010. Inducing tree-substitution grammars. *The Journal of Machine Learning Research* 11 (2010), 3053–3096.

[30] Michael Collins, Jan Hajic, Lance A. Ramshaw, and Christoph Tillmann. 1999. A Statistical Parser for Czech. In *27th Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, Maryland, USA, 20-26 June 1999*, Robert Dale and Kenneth Ward Church (Eds.). ACL, 505–512. https://doi.org/10.3115/1034678.1034754

[31] Robert Dale, Hermann Moisl, and Harold Somers. 2000. Handbook of natural language processing. *Imprint New York: Marcel Dekker* (2000).

[32] Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics* 47, 2 (2021), 255–308.

[33] Marie-Catherine De Marneffe and Joakim Nivre. 2019. Dependency grammar. *Annual Review of Linguistics* 5 (2019), 197–218.

[34] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)* 39, 1 (1977), 1–22.

[35] Andrew Drozdov, Subendhu Rongali, Yi-Pei Chen, Tim O'Gorman, Mohit Iyyer, and Andrew McCallum. 2020. Unsupervised Parsing with S-DIORA: Single Tree Encoding for Deep Inside-Outside Recursive Autoencoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4832–4845. https://doi.org/10.18653/v1/2020.emnlp-main.392

[36] Arianna D'Ulizia, Fernando Ferri, and Patrizia Grifoni. 2011. A survey of grammatical inference methods for natural language learning. *Artif. Intell. Rev.* 36, 1 (2011), 1–27. https://doi.org/10.1007/s10462-010-9199-1

[37] Jason Eisner. 1997. Three New Probabilistic Models for Dependency Parsing: An Exploration. *CoRR* cmp-lg/9706003 (1997). http://arxiv.org/abs/cmp-lg/9706003

[38] Jason Eisner and Giorgio Satta. 1999. Efficient Parsing for Bilexical Context-Free Grammars and Head Automaton Grammars. In *27th Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, Maryland, USA, 20-26 June 1999*, Robert Dale and Kenneth Ward Church (Eds.). ACL, 457–464. https://doi.org/10.3115/1034678.1034748

[39] Hisham El-Shishiny. 1990. A formal description of Arabic syntax in definite clause grammar. In *COLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*.

[40] Vyvyan Evans. 2006. *Cognitive linguistics*. Edinburgh University Press.

[41] G David Forney. 1973. The viterbi algorithm. *Proc. IEEE* 61, 3 (1973), 268–278.

[42] King-Sun Fu and Taylor L. Booth. 1975. Grammatical Inference: Introduction and Survey - Part I. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-5, 1 (1975), 95–111. https://doi.org/10.1109/TSMC.1975.5409159

[43] King-Sun Fu and Taylor L. Booth. 1986. Grammatical Inference: Introduction and Survey-Part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8, 3 (1986), 343–359. https://doi.org/10.1109/TPAMI.1986.4767796

[44] Stuart Geman and Mark Johnson. 2002. Probabilistic grammars and their applications. *International Encyclopedia of the Social & Behavioral Sciences* 2002 (2002), 12075–12082.

[45] Samuel J Gershman and David M Blei. 2012. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology* 56, 1 (2012), 1–12.

[46] Jennifer Gillenwater, Kuzman Ganchev, João Graça, Fernando Pereira, and Ben Taskar. 2010. Sparsity in Dependency Grammar Induction. (2010), 194–199. https://aclanthology.org/P10-2036/

[47] Dave Golland, John DeNero, and Jakob Uszkoreit. 2012. A Feature-Rich Constituent Context Model for Grammar Induction. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*. The Association for Computer Linguistics, 17–22. https://aclanthology.org/P12-2004/

[48] João Graça, Kuzman Ganchev, and Ben Taskar. 2007. Expectation Maximization and Posterior Constraints. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis (Eds.). Curran Associates, Inc., 569–576. https://proceedings.neurips.cc/paper/2007/hash/73e5080f0f3804cb9cf470a8ce895dac-Abstract.html

[49] Peiming Guo, Shen Huang, Peijie Jiang, Yueheng Sun, Meishan Zhang, and Min Zhang. 2022. Curriculum-Style Fine-Grained Adaption for Unsupervised Cross-Lingual Dependency Transfer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2022), 322–332.

[50] Michael Gusenbauer. 2019. Google Scholar to overshadow them all? Comparing the sizes of 12 academic search engines and bibliographic databases. *Scientometrics* (2019), 177–214.

[51] Wenjuan Han, Yong Jiang, Hwee Tou Ng, and Kewei Tu. 2020. A Survey of Unsupervised Dependency Parsing. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 2522–2533. https://doi.org/10.18653/v1/2020.coling-main.227

[52] Wenjuan Han, Yong Jiang, and Kewei Tu. 2017. Dependency Grammar Induction with Neural Lexicalization and Big Training Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 1683–1688. https://doi.org/10.18653/v1/d17-1176

[53] Wenjuan Han, Yong Jiang, and Kewei Tu. 2019. Enhancing unsupervised generative dependency parser with contextual information. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5315–5325.

[54] Randy Allen Harris. 2021. *The linguistics wars: Chomsky, Lakoff, and the battle over deep structure.* Oxford University Press.

[55] Zellig S Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.

[56] Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised Learning of Syntactic Structure with Invertible Neural Projections. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 1292–1302. https://doi.org/10.18653/v1/d18-1160

[57] William P Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the association for computational linguistics*. 101–109.

[58] Kristy Hollingshead, Seeger Fisher, and Brian Roark. 2005. Comparing and Combining Finite-State and Context-Free Parsers. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*. The Association for Computational Linguistics, 787–794. https://aclanthology.org/H05-1099/

[59] Xiang Hu, Haitao Mi, Liang Li, and Gerard de Melo. 2022. Fast-R2D2: A Pretrained Recursive Neural Network based on Pruned CKY for Grammar Induction and Text Representation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 2809–2821. https://doi.org/10.18653/v1/2022.emnlp-main.181

[60] Xiang Hu, Haitao Mi, Zujie Wen, Yafang Wang, Yi Su, Jing Zheng, and Gerard de Melo. 2021. R2D2: Recursive transformer based on differentiable tree for interpretable hierarchical language modeling. *arXiv preprint arXiv:2107.00967* (2021).

[61] Ramon Ferrer i Cancho, Ricard V Solé, and Reinhard Köhler. 2004. Patterns in syntactic dependency networks. *Physical Review E* 69, 5 (2004), 051915.

[62] Samireh Jalali and Claes Wohlin. 2012. Systematic literature studies: database searches vs. backward snowballing. In *2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '12, Lund, Sweden - September 19 - 20, 2012*, Per Runeson, Martin Höst, Emilia Mendes, Anneliese Amschler Andrews, and Rachel Harrison (Eds.). ACM, 29–38. https://doi.org/10.1145/2372251.2372257

[63] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language?. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 3651–3657. https://doi.org/10.18653/v1/p19-1356

[64] Yong Jiang, Wenjuan Han, and Kewei Tu. 2016. Unsupervised neural dependency parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 763–771.

[65] Lifeng Jin, Finale Doshi-Velez, Timothy A. Miller, William Schuler, and Lane Schwartz. 2018. Unsupervised Grammar Induction with Depth-bounded PCFG. *Trans. Assoc. Comput. Linguistics* 6 (2018), 211–224. https://doi.org/10.1162/tacl_a_00016

[66] Lifeng Jin, Finale Doshi-Velez, Timothy A. Miller, Lane Schwartz, and William Schuler. 2019. Unsupervised Learning of PCFGs with Normalizing Flow. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 2442–2452. https://doi.org/10.18653/v1/p19-1234

[67] Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

[68] Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2020. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. *arXiv preprint arXiv:2002.00737* (2020).

[69] Yoon Kim, Chris Dyer, and Alexander M. Rush. 2019. Compound Probabilistic Context-Free Grammars for Grammar Induction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, Anna Korhonen, David R. Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, 2369–2385. https://doi.org/10.18653/v1/p19-1228

[70] Barbara A. Kitchenham and Pearl Brereton. 2013. A systematic review of systematic review process research in software engineering. *Inf. Softw. Technol.* 55, 12 (2013), 2049–2075. https://doi.org/10.1016/j.infsof.2013.07.010

[71] Dan Klein and Christopher D. Manning. 2002. A Generative Constituent-Context Model for Improved Grammar Induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 128–135. https://doi.org/10.3115/1073083.1073106

[72] Dan Klein and Christopher D. Manning. 2004. Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, Donia Scott, Walter Daelemans, and Marilyn A. Walker (Eds.). ACL, 478–485. https://doi.org/10.3115/1218955.1219016

[73] Tamara G. Kolda and Brett W. Bader. 2009. Tensor Decompositions and Applications. *SIAM Rev.* 51, 3 (2009), 455–500. https://doi.org/10.1137/07070111X

[74] Marco Kuhlmann. 2010. *Dependency Structures and Lexicalized Grammars An Algebraic Approach*. Lecture Notes in Computer Science, Vol. 6270. Springer. https://doi.org/10.1007/978-3-642-14568-1

[75] G LAKOFF. 1987. Woman, Fire, and Dangerous Things. *What Categories Reveal about the Mind* (1987).

[76] Ronald W Langacker. 1987. *Foundations of cognitive grammar: Volume I: Theoretical prerequisites*. Vol. 1. Stanford university press.

[77] Ronald W Langacker. 1995. Structural syntax: the view from cognitive grammar. *Dondelinger et al.[1995]* (1995), 13–37.

[78] Phong Le and Willem H. Zuidema. 2014. The Inside-Outside Recursive Neural Network model for Dependency Parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 729–739. https://doi.org/10.3115/v1/d14-1081

[79] Phong Le and Willem H. Zuidema. 2015. Unsupervised Dependency Parsing: Let's Use Supervised Parsers. (2015), 651–661. https://doi.org/10.3115/v1/n15-1067

[80] Boda Lin, Zijun Yao, Jiaxin Shi, Shulin Cao, Binghao Tang, Si Li, Yong Luo, Juanzi Li, and Lei Hou. 2022. Dependency Parsing via Sequence Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022,December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 7339–7353. https://doi.org/10.18653/v1/2022.findings-emnlp.543

[81] DC Liu and J Nocedal. 1989. On the limited memory method for large scale optimization: Mathematical Programming B. (1989).

[82] Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews* 21 (2017), 171–193.

[83] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[84] Markéta Lopatková, Martin Plátek, and Vladislav Kubon. 2005. Modeling Syntax of Free Word-Order Languages: Dependency Analysis by Reduction. In *Text, Speech and Dialogue, 8th International Conference, TSD 2005, Karlovy Vary, Czech Republic, September 12-15, 2005, Proceedings (Lecture Notes in Computer Science, Vol. 3658)*, Václav Matousek, Pavel Mautner, and Tomás Pavelka (Eds.). Springer, 140–147. https://doi.org/10.1007/11551874_18

[85] David Luengo, Luca Martino, Mónica F. Bugallo, Víctor Elvira, and Simo Särkkä. 2020. A survey of Monte Carlo methods for parameter estimation. *EURASIP J. Adv. Signal Process.* 2020, 1 (2020), 25. https://doi.org/10.1186/s13634-020-00675-6
[86] Hongyin Luo, Lan Jiang, Yonatan Belinkov, and James Glass. 2019. Improving neural language models by segmenting, attending, and predicting the future. *arXiv preprint arXiv:1906.01702* (2019).
[87] Brian MacWhinney. 2000. The CHILDES project: Tools for analyzing talk: Volume I: Transcription format and programs, volume II: The database.
[88] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguistics* 19, 2 (1993), 313–330.
[89] David Marecek. 2016. Twelve Years of Unsupervised Dependency Parsing.. In *ITAT*. 56–62.
[90] David Marecek and Milan Straka. 2013. Stop-probability estimates computed on a large corpus improve Unsupervised Dependency Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*. The Association for Computer Linguistics, 281–290. https://aclanthology.org/P13-1028/
[91] David Mareček and Zdeněk Žabokrtskỳ. 2011. Gibbs sampling with treeness constraint in unsupervised dependency parsing. In *Proceedings of Workshop on Robust Unsupervised and Semisupervised Methods in Natural Language Processing*. 1–8.
[92] David Mareček and Zdeněk Žabokrtskỳ. 2012. Exploiting reducibility in unsupervised dependency parsing. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 297–307.
[93] Stephanos Matthaios. 2011. Eratosthenes of Cyrene: Readings of his 'Grammar'definition. *Ancient Scholarship and Grammar: Archetypes, Concepts and Contexts. Trends in classics-supplementary volumes* 8 (2011), 55–86.
[94] Igor Aleksandrovic Mel'cuk et al. 1988. *Dependency syntax: theory and practice*. SUNY press.
[95] Ruslan Mitkov. 2022. *The Oxford handbook of computational linguistics*. Oxford University Press.
[96] Vigneshwaran Muralidaran, Irena Spasić, and Dawn Knight. 2021. A systematic review of unsupervised approaches to grammar induction. *Natural Language Engineering* 27, 6 (2021), 647–689.
[97] Hiroshi Noji, Yusuke Miyao, and Mark Johnson. 2016. Using Left-corner Parsing to Encode Universal Structural Constraints in Grammar Induction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, Jian Su, Xavier Carreras, and Kevin Duh (Eds.). The Association for Computational Linguistics, 33–43. https://doi.org/10.18653/v1/d16-1004
[98] Ankur Parikh, Shay B Cohen, and Eric Xing. 2014. Spectral unsupervised parsing with additive tree metrics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1062–1072.
[99] John K. Pate and Sharon Goldwater. 2013. Unsupervised Dependency Parsing with Acoustic Cues. *Trans. Assoc. Comput. Linguistics* 1 (2013), 63–74. https://doi.org/10.1162/tacl_a_00210
[100] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. https://doi.org/10.3115/v1/d14-1162
[101] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inf. Softw. Technol.* 64 (2015), 1–18. https://doi.org/10.1016/j.infsof.2015.03.007
[102] Jim Pitman. 1995. Exchangeable and partially exchangeable random partitions. *Probability theory and related fields* 102, 2 (1995), 145–158.
[103] Jim Pitman and Marc Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability* (1997), 855–900.
[104] Elias Ponvert, Jason Baldridge, and Katrin Erk. 2011. Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 1077–1086.
[105] Ines Rehbein and Josef van Genabith. 2007. Evaluating Evaluation Measures. In *Proceedings of the 16th Nordic Conference of Computational Linguistics, NODALIDA 2007, Tartu, Estonia, May 2007*, Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek, and Mare Koit (Eds.). University of Tartu, Estonia, 372–379. https://aclanthology.org/W07-2460/
[106] Philip Resnik. 1992. Left-corner parsing and psychological plausibility. In *COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics*.
[107] Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*, Francis R. Bach and David M. Blei (Eds.). JMLR.org, 1530–1538. http://proceedings.mlr.press/v37/rezende15.html
[108] Herbert Robbins. 1951. Asymptotically subminimax solutions of compound statistical decision problems. In *Proceedings of the second Berkeley symposium on mathematical statistics and probability*, Vol. 2. University of California Press, 131–149.
[109] Rudolf Rosa and David Marecek. 2019. Inducing Syntactic Trees from BERT Representations. *CoRR* abs/1906.11511 (2019). arXiv:1906.11511 http://arxiv.org/abs/1906.11511

[110] Rajhans Samdani, Ming-Wei Chang, and Dan Roth. 2012. Unified Expectation Maximization. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*. The Association for Computational Linguistics, 688–698. https://aclanthology.org/N12-1087/

[111] Baskaran Sankaran. 2010. A survey of unsupervised grammar induction. (2010).

[112] Jesus Santamaria and Lourdes Araujo. 2010. Identifying patterns for unsupervised grammar induction. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. 38–45.

[113] Yoav Seginer. 2007. Fast Unsupervised Incremental Parsing. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, John Carroll, Antal van den Bosch, and Annie Zaenen (Eds.). The Association for Computational Linguistics. https://aclanthology.org/P07-1049/

[114] Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. 2018. Neural Language Modeling by Jointly Learning Syntax and Lexicon. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rkgOLb-0W

[115] Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron C. Courville. 2021. StructFormer: Joint Unsupervised Induction of Dependency and Constituency Structure from Masked Language Modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 7196–7209. https://doi.org/10.18653/v1/2021.acl-long.559

[116] Khalil Sima'an. 1996. Computational Complexity of Probabilistic Disambiguation by means of Tree-Grammars. In *16th International Conference on Computational Linguistics, Proceedings of the Conference, COLING 1996, Center for Sprogteknologi, Copenhagen, Denmark, August 5-9, 1996*. 1175–1180. https://aclanthology.org/C96-2215/

[117] Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A linguistically interpreted corpus of German newspaper text. *arXiv preprint cmp-lg/9807008* (1998).

[118] Kenny Smith, Henry Brighton, and Simon Kirby. 2003. Complex Systems in Language Evolution: the Cultural Emergence of Compositional Structure. *Adv. Complex Syst.* 6, 4 (2003), 537–558. https://doi.org/10.1142/S0219525903001055

[119] Noah A Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 354–362.

[120] Noah A Smith and Jason Eisner. 2005. Guiding unsupervised grammar induction using contrastive estimation. In *Proc. of IJCAI Workshop on Grammatical Inference Applications*. 73–82.

[121] Noah A Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. 569–576.

[122] Noah A. Smith and Mark Johnson. 2007. Weighted and Probabilistic Context-Free Grammars Are Equally Expressive. *Comput. Linguistics* 33, 4 (2007), 477–491. https://doi.org/10.1162/coli.2007.33.4.477

[123] Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. 2005. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences* 102, 33 (2005), 11629–11634.

[124] Valentin I. Spitkovsky, Hiyan Alshawi, Angel X. Chang, and Daniel Jurafsky. 2011. Unsupervised Dependency Parsing without Gold Part-of-Speech Tags. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1281–1290. https://aclanthology.org/D11-1118/

[125] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From Baby Steps to Leapfrog: How "Less is More" in Unsupervised Dependency Parsing. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*. The Association for Computational Linguistics, 751–759. https://aclanthology.org/N10-1116/

[126] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2011. Lateen EM: Unsupervised Training with Multiple Objectives, Applied to Dependency Grammar Induction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1269–1280. https://aclanthology.org/D11-1117/

[127] Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2011. Punctuation: Making a point in unsupervised dependency parsing. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. 19–28.

[128] Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2012. Capitalization cues improve dependency grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*. 16–22.

[129] Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2012. Three dependency-and-boundary models for grammar induction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 688–698.

[130] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2013. Breaking Out of Local Optima with Count Transforms and Model Recombination: A Study in Grammar Induction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1983–1995. https://aclanthology.org/D13-1204/

[131] Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. 2010. Viterbi Training Improves Unsupervised Dependency Parsing. (2010), 9–17. https://aclanthology.org/W10-2902/

[132] Mark Steedman. 2001. *The syntactic process*. MIT press.

[133] Mark Steedman and Jason Baldridge. 2011. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell* (2011), 181–224.

[134] Andrew Stevenson and James R Cordy. 2014. A survey of grammatical inference in software engineering. *Science of Computer Programming* 96 (2014), 444–459.

[135] Andreas Stolcke and Stephen Omohundro. 1994. Inducing probabilistic grammars by Bayesian model merging. In *International Colloquium on Grammatical Inference*. Springer, 106–118.

[136] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27 (2014).

[137] Yee Whye Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. 985–992.

[138] Lucien Tesnière. 1959. Éléments de syntaxe structurale. (1959).

[139] Michael Tomasello. 2000. Do young children have adult syntactic competence? *Cognition* 74, 3 (2000), 209–253.

[140] Kewei Tu and Vasant Honavar. 2012. Unambiguity regularization for unsupervised learning of probabilistic grammars. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1324–1334.

[141] Kewei Tu, Yong Jiang, Wenjuan Han, and Yanpeng Zhao. 2021. Unsupervised Natural Language Parsing (Introductory Tutorial). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*. 1–5.

[142] Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in ATIS?. In *2010 IEEE Spoken Language Technology Workshop*. 19–24. https://doi.org/10.1109/SLT.2010.5700816

[143] Marten van Schijndel, Andy Exley, and William Schuler. 2013. A model of language processing as hierarchic sequential prediction. *Topics in cognitive science* 5, 3 (2013), 522–540.

[144] Menno van Zaanen. 2001. ABL: Alignment-Based Learning. *CoRR* cs.LG/0104006 (2001). https://arxiv.org/abs/cs/0104006

[145] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[146] Joseph Weizenbaum. 1966. ELIZA - a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (1966), 36–45. https://doi.org/10.1145/365153.365168

[147] Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering* 11, 2 (2005), 207–238.

[148] Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *Proceedings of the Eighth International Conference on Parsing Technologies, IWPT 2003, Nancy, France, April 2003*. 195–206. https://aclanthology.org/W03-3023/

[149] Erjia Yan and Ying Ding. 2010. Applying centrality measures to impact analysis: A coauthorship network analysis. *CoRR* abs/1012.4862 (2010). arXiv:1012.4862 http://arxiv.org/abs/1012.4862

[150] Songlin Yang, Yong Jiang, Wenjuan Han, and Kewei Tu. 2020. Second-Order Unsupervised Neural Dependency Parsing. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, Donia Scott, Núria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, 3911–3924. https://doi.org/10.18653/v1/2020.coling-main.347

[151] Songlin Yang, Yanpeng Zhao, and Kewei Tu. 2021. Neural Bi-Lexicalized PCFG Induction. (2021), 2688–2699. https://doi.org/10.18653/v1/2021.acl-long.209

[152] Songlin Yang, Yanpeng Zhao, and Kewei Tu. 2021. PCFGs Can Do Better: Inducing Probabilistic Context-Free Grammars with Many Symbols. (2021), 1487–1498. https://doi.org/10.18653/v1/2021.naacl-main.117

[153] Zhiyuan Zeng and Deyi Xiong. 2022. Unsupervised and few-shot parsing from pretrained language models. *Artif. Intell.* 305 (2022), 103665. https://doi.org/10.1016/j.artint.2022.103665

[154] Hao Zhu, Yonatan Bisk, and Graham Neubig. 2020. The return of lexical dependencies: Neural lexicalized PCFGs. *Transactions of the Association for Computational Linguistics* 8 (2020), 647–661.

CAPÍTULO

4

INDUÇÃO GRAMATICAL PARA O PORTUGUÊS: A CONTRIBUIÇÃO DA INFORMAÇÃO MÚTUA PARA A DESCOBERTA DE RELAÇÕES DE DEPENDÊNCIA

O artigo intitulado *"Indução Gramatical para o Português: a Contribuição da Informação Mútua para a Descoberta de Relações de Dependência"* teve como autores *Diego Pedro Gonçalves da Silva* e *Thiago Alexandre Salgueiro Pardo*. O trabalho foi publicado na Jornada de Descrição do Português, evento satélite do *Symposium in Information and Human Language Technology*–***STIL*** no ano de 2023. Este trabalho foi motivado pelos achados do estudo produzido por Futrell *et al.* (2019) onde se apresentam evidências da influência de informação mutual na identificação de relações de independência na língua inglesa. Devido a falta de estudos em língua portuguesa, foi investigado se o mesmo padrão para a língua inglesa era apresentado na língua portuguesa, mesmo usando um córpus muito menor. As principais contribuições relevantes para esta dissertação residem na descoberta da similaridade de performance em línguas distintas. Além disso, foi identificado que algumas relações sintáticas apresentam maior IM que outras relações sintáticas.

Indução Gramatical para o Português: a Contribuição da Informação Mútua para Descoberta de Relações de Dependência

Diego Pedro Gonçalves da Silva[1], **Thiago Alexandre Salgueiro Pardo**[1]

[1]Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo

diegopedro@usp.br, taspardo@icmc.usp.br

Resumo. *Indução gramatical é uma tarefa que busca aprender automaticamente estruturas sintáticas a partir de texto. Poucos trabalhos de indução gramatical foram produzidos direcionados para a língua portuguesa. Neste artigo, reproduzidos o trabalho de [Futrell et al. 2019] para a língua portuguesa e o estendemos ao incluir análise de informação mútua para relações sintáticas específicas. Utilizamos dois treebanks anotados e realizamos experimentos utilizando embeddings de dimensões variadas, demonstrando a hipótese de alta informação mútua para palavras em relações de dependência.*

1. Introdução

Na Linguística, sintaxe é definida como o estudo da organização das palavras (em termos de ordenação e estruturação) na formação de sentenças. Esse entendimento é compartilhado por diferentes visões sobre como a sintaxe deve ser formalizada [Chomsky 2014] [Bresnan et al. 2015] . Quase toda aplicação de Processamento de Línguas Naturais (PLN) necessita de algum conhecimento sintático para obter bons resultados, direta ou indiretamente codificados. Revisores gramaticais, sistemas de simplificação de textos e sistemas de extração de informação são algumas das aplicações que se beneficiam da representação explícita da sintaxe. As aplicações baseadas em grandes modelos de língua, por sua vez, acabam adquirindo noções de sintaxe em seu treinamento, mesmo que ela não seja completamente explicitada..

Dada a relevância da sintaxe, a Indução Gramatical (IG), também chamada de *parsing* não supervisionado [Klein e Manning 2004], é uma tarefa de interesse na comunidade de PLN. Apesar de ela ter a finalidade de induzir ("aprender") automaticamente a gramática a partir de dados textuais sem anotações sintáticas [Klein e Manning 2004], vários autores realizam IG como tarefa semi-supervisionada (IGSS) ou supervisionada (IGS) [Headden III et al. 2009] [Spitkovsky et al. 2013]. Sendo assim, utilizaremos o termo (IGNS) para nos referirmos à tarefa de IG não supervisionada. É interessante notar que, independentemente de aplicações computacionais de PLN, a IGNS pode auxiliar em várias frentes. Na Linguística, pode ser útil para aprender a gramática de línguas mortas ou com escassez de recursos (como as indígenas) [Dahl et al. 2023]. Em Psicolinguística, pode ser utilizada para propor modelos de aquisição da linguagem [Bannard et al. 2009]. Em Bioinformática, IGNS é utilizada para inferir estruturas de DNA desconhecidas ou difíceis de serem encontradas em grandes bases de dados [Unold et al. 2020].

A sintaxe (e, por consequência, a IGNS) se vale de duas visões diferentes de representação: a gramática de constituinte e a gramática de dependência. A primeira

estuda como as sentenças são formadas por blocos básicos (sintagmas). No modelo de representação de gramática de dependência, foco desse artigo, estabelecem-se relações de dependências diretamente entre as palavras. A Figura 1 apresenta relações de sujeito (nsubj) (entre o verbo "chutou" e a palavra "menino") e objeto (obj) (entre o verbo e a palavra "bola"), por exemplo.

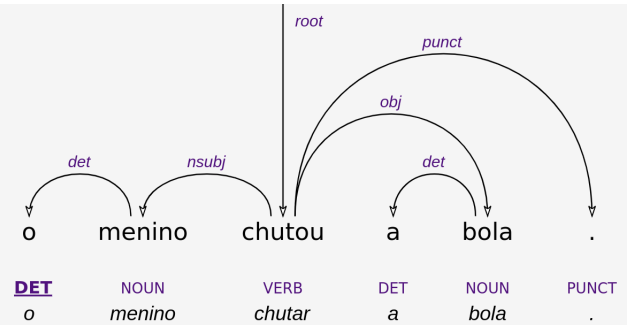


Figura 1. Exemplo de análise de dependência

Devido aos avanços da IGNS nos anos 2000, presumia-se que a IGNS se aproximaria da IGS em desempenho em breve, mas isso ainda não ocorreu [Bod 2007] [Lin et al. 2022]. A maioria dos trabalhos desenvolvidos nas últimas décadas utiliza algum tipo de anotação. Induzir gramática sem nenhuma informação prévia é uma tarefa bastante difícil. Outro desafio é a indução de gramática em sentenças independentemente do tamanho. A maioria dos trabalhos publicados utiliza sentenças de até 10 palavras. Por fim, as diferenças sintáticas entre as línguas apresentam mais um desafio que dificulta a padronização de técnicas para diferentes línguas. Por exemplo, as línguas chinesa, persa e tupi apresentam características linguísticas diferentes por fazerem parte de famílias de línguas diferentes [Theodor e Siebert-Cole 2020], o que pode dificultar a criação de um modelo unificado. No melhor de nosso conhecimento, não encontramos trabalhos publicados de IGNS específico para a língua portuguesa. O trabalho mais similar encontrado foi produzido por [da Costa e Kepler 2014], que implementa uma abordagem semi-supervisionada baseado no trabalho de [Klein e Manning 2004].

Nesse contexto, nosso objetivo neste artigo é explorar a tarefa de IGNS para o português. Em específico, focamos na reprodução de um experimento recente de uso da conhecida medida de Informação Mútua (IM) para tentar predizer palavras que possam estar relacionadas sintaticamente. A IM é uma medida de dependência, assim, quanto maior a informação mútua entre duas palavras, maior a chance de elas estarem relacionadas. Baseamo-nos no trabalho de [Futrell et al. 2019], que, usando IM aplicada a um corpus de milhões de palavras em inglês, mostrou que há uma maior IM entre palavras que mantêm relação de dependência do que entre palavras que não mantêm. Além de avaliar tal técnica para o português, vamos além e verificamos seu comportamento para relações específicas. Realizamos nossos experimentos com dados de *treebanks* alinhados ao modelo *Universal Dependencies* (UD) [de Marneffe et al. 2021], amplamente adotado.

Na Seção 2, apresentamos brevemente os principais trabalhos relacionados. Em seguida, na Seção 3, descrevemos a abordagem aplicada no nosso estudo. Na Seção 4, apresentamos os resultados do estudo. Fazemos algumas considerações finais na Seção 5.

2. Trabalhos relacionados

Ao longo das últimas décadas, várias abordagens foram utilizadas em IGNS. A maioria dos trabalhos utilizam a abordagem gerativa, principalmente no uso do algoritmo *Expectation–maximization – EM* [Baker 1979], que é utilizado para estimar a probabilidade de variáveis não observáveis (árvores sintáticas em IGNS). Nos últimos anos, a modelagem neural vem ganhando bastante espaço.

Ao longo das duas últimas décadas, o modelo DMV (*Dependency Model with Valence*) [Klein e Manning 2004] exerceu grande influência para gramática de dependência. A ideia por trás do modelo DMV está no controle de geração da árvore sintática, que, para cada ramo a ser gerado (relação de dependência), utiliza-se de distribuições de probabilidade para tomar decisões de quando gerar ($P_{STOP}(\neg STOP|h, dir, adj)$) e qual ramo gerar ($P_{CHOOSE}(a|h, dir)$). As variáveis *h*, *dir*, *a* e *dij* são respectivamente a cabeça da relação, a direção em que o argumento será gerado (direita ou esquerda), o argumento a ser gerado e se o argumento já foi gerado na árvore na direção *dir*. O DMV é um dos vários modelos que utilizam o EM. Este foi o primeiro trabalho a ultrapassar o *baseline* de ramificação direita (*right-branching*) [Headden III et al. 2009], sendo bastante utilizado, mesmo com quase duas décadas de existência [Yang et al. 2020].

Muitos trabalhos foram influenciados por [Klein e Manning 2004]. Um dos mais relevantes, [Headden III et al. 2009] estendeu o modelo DMV para aplicar uma abordagem Bayesiana, em vez de EM, utilizando uma gramática lexicalizada (cada nó da árvore sintática contém também informação sobre o léxico a que se refere). [Cohen e Smith 2009] optou por substituir a distribuição *Dirichlet* pela Logística, pois, apesar de a primeira ser mais fácil de treinar, ela não permite um meio explícito de forma flexível para calcular a covariância entre dois eventos, conforme descreve [Blei e Lafferty 2005]. O trabalho alcançou 42% de *Direct Dependency Accuracy – DDA* (quando considera a direção de geração da árvore sintática) no corpus WSJ∞, para sentenças de qualquer tamanho.

O trabalho de [Spitkovsky et al. 2010] obteve bons resultados a partir da aplicação de *curriculum learning* [Bengio et al. 2009], que inicia o treinamento com dados menos complexos e aumenta a complexidade dos dados até que toda a base de dados tenha sido utilizada. A mudança de complexidade contribui para que se reduzam as chances de cair em máximos locais (um dos problemas no uso de EM usado para problemas não convexos). Este trabalho obteve 45% de DDA no WSJ∞. Mais recentemente, [Han et al. 2019b] propôs o *Lexicalized Neural Dependency Model with Valence* (L-NDMV), um modelo lexicalizado que utiliza DMV com redes neurais. Esse trabalho constatou que, ao explorar características lexicais, a tarefa de IGNS ganha em desempenho. O L-NDMV foi o primeiro trabalho a ultrapassar a marca dos 60% de DDA no WSJ∞, enquanto que os trabalhos supervisionados ultrapassam a marca dos 95% em *Unlabeled Attachment Score – UAS* (quando não considera a direção de geração da árvore) [Lin et al. 2022].

[Yang et al. 2020] atingiu o estado da arte ao construir o modelo probabilístico com mais de um nível de distância de hierarquia (além de *filhos*, *pais* e *irmãos* também considera *avôs*, *netos* e *tios*, por exemplo) entre os nós da árvore. Outros trabalhos atingiram o estado da arte ao estender o modelo DMV com redes neurais [Han et al. 2019a] [Han et al. 2017] [Jiang et al. 2016]. Todos estes trabalhos utilizam algum tipo de informação léxica e redes neurais para contribuir com o desempenho. [Shen et al. 2021] usa o conceito de distância e altura sintática para segmentar a sentença em partes menores.

[Drozdov et al. 2019] aplica o algoritmo *Inside-Outside – IO*, que pode ser visto como uma instância do EM, em redes neurais.

Todos os trabalhos citados utilizam algum tipo de anotação no treinamento. Recentemente, [Pate e Johnson 2016] treinou o modelo DMV com milhões de palavras para induzir dependência sem uso de anotação. Uma vez que o modelo não utiliza categorias morfossintáticas, é utilizada inferência Bayesiana aplicada a gramáticas livres de contexto probabilísticas. Apesar de já existirem estudos anteriores que utilizavam apenas palavras como entrada para o modelo [Seginer 2007], estes eram apenas para constituintes.

O uso de IM, em específico, é algo que vem sendo relativamente pouco explorado, apesar de seu claro apelo para a tarefa. [Magerman e Marcus 1990] foi o primeiro trabalho a aplicar IM em IGS, mas foi recentemente que IM começou a ser aplicado em tarefas de IGNS. Em um trabalho recente, [Futrell et al. 2019] constatou que pares de palavras que têm relação sintática apresentam uma maior IM quando comparados a pares de palavras sem relação. Esta hipótese também foi aplicada em indução gramatical por [Hoover et al. 2021], que utilizou o modelo de língua pré-treinado para calcular informação mútua entre palavras considerando o contexto.

A seguir, detalhamos o método de [Futrell et al. 2019] e como o reproduzimos.

3. Método de indução gramatical

O trabalho de [Futrell et al. 2019] utilizou um corpus com 320 milhões de *tokens* anotados automaticamente. Os autores analisam três variáveis: palavras (*words*), categorias morfossintáticas (*pos*) e grupos lexicais (*lex*). A última variável é resultante de agrupamento. O trabalho propôs um agrupamento com os 60K *tokens* mais frequentes, incluindo *stopwords* e pontuação, a fim de ter uma dimensionalidade menor. Utilizando os vetores de 300 dimensões do modelo *Glove* para cada uma das 60K palavras, o trabalho agrupa os tokens em 300 grupos. Por exemplo, os *tokens* "carro", "carros", "automóvel" e "automóveis" fazem parte do mesmo grupo lexical. Para representar a variável *lex*, cada *token* no corpus é substituído pelo número do seu respectivo grupo.

A IM é calculada entre pares de palavras, categorias morfossintáticas e grupos lexicais. Para pares de palavras, por exemplo, na sentença "O menino chutou a bola", alguns dos pares possíveis são <o,chutou> e <chutou,bola>. O primeiro par não tem relação de dependência (indicada no experimento como *nondep*). O segundo par tem relação, conforme apresentado na Figura 1 (indicada como *dep*). [Futrell et al. 2019] quis também saber o desempenho de pares aleatórios. Ele descreveu estes pares como *permuted*. O mesmo é estabelecido para categorias morfossintáticas e grupos lexicais. Ao todo, para cada variável, 3 experimentos são realizados. O cálculo da IM ocorre entre o termo cabeça da relação *h* e o dependente *d*, cuja fórmula é apresentada abaixo. A fórmula calcula a probabilidade de haver uma relação entre duas variáveis, que podem ser palavras, categorias morfossintáticas, grupos lexicais e relações sintáticas (usadas neste trabalho).

$$IM = log \frac{P(h,d)}{P(h)P(d)}$$

Para avaliar seu modelo, [Futrell et al. 2019] utiliza dois *baselines*: pares permutados (*words perm*, *lex perm*, *pos perm*) e pares sem relação de dependência (*words nondep*,

lex nondep, *pos nondep*). [Futrell et al. 2019] utilizou estes dois baselines porque o primeiro considera uma relação aleatória na sentença, podendo ser de dependência (como em <o,menino>) ou não (<o,a>). Assim, o *baseline* permutado deve apresentar melhor desempenho do que o *baseline* sem dependência se existir maior informação mútua entre relações de dependência do que não dependência. Estes *baselines* são comparados com os resultados das 3 variáveis para relação de dependência (*words dep*, *lex dep*, *pos dep*).

Diferentemente de [Futrell et al. 2019], utilizamos dois corpora anotados por humanos, disponíveis na página do projeto UD: Bosque [Afonso et al. 2002] e Petrogold [de Souza et al. 2021]. Para o Bosque, as sentenças anotadas com Português do Brasil correspondem a 90K *tokens* e 4.205 sentenças originárias do CETENFolha [Linguateca 2023], construído com textos jornalísticos. O corpus PetroGold é formado por 19 teses e dissertações na área de óleo e gás, constituído por 232k *tokens* e 9k sentenças. O corpus Bosque contém 4,16% das sentenças com até 3 *tokens* e 8,3% das sentenças com mais de 40 *tokens*. O PetroGold contém 4,5% das sentenças com até 3 palavras e 24,7% com sentenças acima de 40 *tokens*. Além da distribuição diferente de tamanho de sentença, constatamos que os corpora apresentam também diferenças na distribuição de categorias morfossintáticas e funções sintáticas, apesar de essas diferenças serem pequenas.

Adotamos o método de [Futrell et al. 2019], mas utilizamos apenas anotações que foram produzidas por humanos, sem análise automática. Além disso, não selecionamos os *tokens* mais frequentes, uma vez que o tamanho do vocabulário dos corpora compilados é menor que o tamanho proposto por [Futrell et al. 2019]. Em vez disso, apenas utilizamos todo o vocabulário dos corpora que é representado no modelo Glove treinado para a língua portuguesa [Hartmann et al. 2017], totalizando um vocabulário de 21.428 palavras.

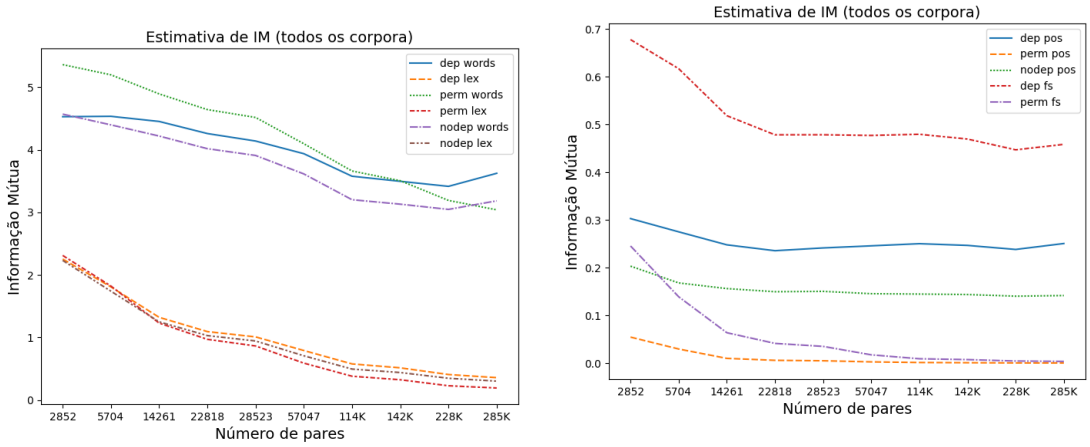
Neste trabalho, utilizamos *embeddings* de 50, 300 e 600 dimensões com base na análise realizada por [Hartmann et al. 2017] para gerar os grupos lexicais. Para definir os 300 grupos, [Futrell et al. 2019] utiliza uma matriz de similaridade. No entanto, [Futrell et al. 2019] não informa como essa matriz de similaridade foi construída. Deduzimos que a matriz foi construída usando similaridade de cosseno entre as *embeddings* de cada *token*. Devido às limitações computacionais, foram utilizadas apenas duas casas decimais para representar os valores na matriz de similaridade.

Além dos dois corpora isoladamente, também usamos a combinação deles. Para cada corpus, foram realizadas 3 execuções, uma para cada uma das 3 dimensões, contabilizando um total de 9 execuções. O código utilizado para a realização dos experimentos foi o mesmo disponibilizado por [Futrell et al. 2019].

4. Resultados

Na Figura 2, são apresentados os gráficos com os resultados para IM utilizando todos os corpora. Resolvemos utilizar os nomes originais das variáveis utilizados no trabalho do [Futrell et al. 2019] para facilitar a reprodução do estudo. Incluímos a variável *fs*, que representa as relações de dependência. Para o agrupamento, variável *lex*, foram utilizadas *embeddings* de 300 dimensões.

Na Figura 2(a), observa-se que, conforme o número de pares aumenta, decresce a informação mútua em todos os grupos, sem tendência de convergência para um valor específico. Este mesmo comportamento é observado no trabalho de [Futrell et al. 2019]

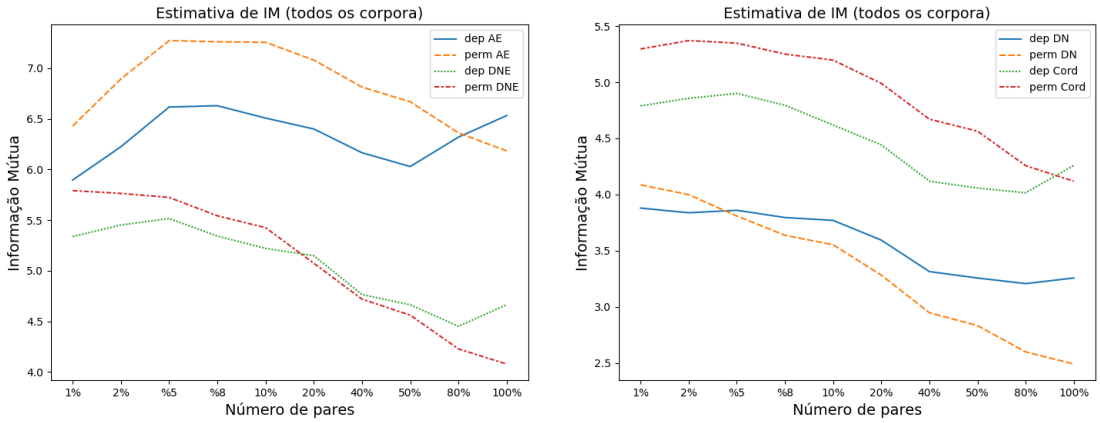


(a) IM de palavras e grupos léxico

(b) IM de grupos e categorias morfossintáticas (pos)

Figura 2. IM por número de pares usando 300 dimensões com todos os corpora

para a língua inglesa, mesmo utilizando um corpus dezenas de vezes maior que o nosso. Isso sugere que a IM entre pares que contenham relações de dependência pode seguir o mesmo padrão para diferentes línguas. Na Figura 2(b), para as variáveis *pos*, observa-se uma estabilidade, também apresentando comportamento similar ao trabalho de [Futrell et al. 2019]. Observa-se também uma IM maior para relações sintáticas do que para categorias morfossintáticas, mesmo com maior esparsidade no grupo de relações sintáticas.



(a) IM de relações sintáticas AE e DNE

(b) IM de relações sintáticas DN e Coord

Figura 3. IM por porcentagem dos pares utilizados com todos os corpora

Na Figura 3, são apresentados os resultados para IM de relações sintáticas separadas pelos grupos definidos pela UD: Argumentos Essenciais (AE) (*Core arguments*, que incluem, por exemplo, as relações sintáticas mais importantes da sentença, como sujeito e objetos), Dependentes Não Essenciais (DNE) (*Non-core dependents*, que incluem, por exemplo, relações de vocativo, modificadores adverbiais e verbos auxiliares), Dependentes Nominais (DN) (*Nominal dependents*, que incluem, por exemplo, modificadores de substantivos e de adjetivos) e coordenações (Coord) (*Coordination*, que incluem, por exemplo, fenômenos variados, como a coordenação por conjunções, expressões multipa-

lavra e relações especiais). Na Figura 3, podemos observar que, dos quatro grupos, AE apresenta a maior IM, assim como também é o único que não apresenta uma tendência de queda conforme o número de pares aumenta.

Durante os experimentos, observamos que o número de dimensões influencia na IM. Nos experimentos usando agrupamento, percebemos que, quanto maior o número de dimensões, menor será a IM, apesar de constatarmos um pequeno aumento de pouco mais de 5% da IM utilizando *embeddings* de 600 dimensões em comparação com *embeddings* de 300 dimensões. Não temos certeza do que pode ter causado esta variação, mas, uma vez que ocorreu uma redução de 50% de IM entre os grupos que utilizaram *embeddings* de 50 e 300, acreditamos que o resultado pode ser devido à alguma característica intrínseca aos *embeddings* utilizados. Não conseguimos identificar uma relação entre o tamanho da sentença e a IM. Um resumo dos resultados para cada corpus é apresentado na Tabela 1. Os dados sugerem que, quanto menor o número de pares, maior a informação mútua.

Tabela 1. Resumo das execuções para todos os corpora

Corpora	$\mu(\sigma)$ **palavras / sentença**	**Número de pares**	**IM dep words**	**IM nondep words**	**IM dep lex**	**IM nondep lex**
Bosque	21,5 (13,51)	70.938	4,638	4,016	0,352	0,301
PetroGold	30,0 (19,5)	221.987	3,275	2,893	0,213	0,187

Finalmente, realizamos experimentos usando as relações sintáticas (Tabela 2). Devido à diferença no número de pares entre as relações, realizamos os experimentos considerando 4 das relações mais relevantes: nsubj, obj, iobj e xcomp.

Tabela 2. Experimento utilizando informações sintáticas

Relações	**Número de pares**	**IM dep words**	**IM permuted**	**IM dep fs**	σ **fs**
nsubj	55.412	**7,488**	0,014	0,478	0,0057
obj	34.692	7,295	0,020	0,631	0.0061
iobj	677	5,026	**0,520**	**1,232**	**0,0159**
xcomp	8.357	5,897	0,088	0,940	0,0122

Os resultados apresentados na Tabela 2 demonstram que há uma diferença muito grande de IM entre relações de dependência e IM com relações permutadas. A relação *iobj* apresenta o melhor desempenho entre as demais relações quando se observa a direção da relação (IM de 1,232), provavelmente devido ao número pequeno de exemplos analisados, uma vez que a IM permutada é bastante alta e o desvio padrão também. No caso da relação *nsubj*, a terceira coluna representa a IM das palavras que fazem parte dessa relação de dependência (sem considerar a direção da relação). Percebe-se uma alta IM nesta categoria, provavelmente devido às características sintáticas do *nsubj*. Como ilustração de pares com alta IM, as maiores IM encontradas para *obj* foram para os pares **computadores – comprei** (com valor 0,03197), **anos – há** (0,01870) e **-se – trata** (0,01336).

5. Considerações finais

Reproduzimos o trabalho de [Futrell et al. 2019] usando corpora da língua portuguesa. Apesar de o tamanho dos corpora usados por [Futrell et al. 2019] e os usados neste trabalho serem bem diferentes, constatamos tendência de comportamento similares, com algumas pequenas diferenças, sugerindo que existe um padrão de comportamento mesmo em línguas pertencentes à famílias linguísticas diferentes. Diferentemente do estudo publicado por [Futrell et al. 2019], que anotou milhões de palavras automaticamente, utilizamos apenas anotações sintáticas de referência produzidas por humanos. Essas variações tornam inconclusivas comparações diretas entre os trabalhos. Além disso, [Futrell et al. 2019] não informa como foi construída a matriz de similaridades.

Trabalhos futuros incluem aplicar este experimento a outros corpora anotados, como o Porttinari [Pardo et al. 2021], e realizar um estudo mais aprofundado sobre a influência nos resultados dos corpora, assim como das *embeddings* utilizadas.

Agradecimentos
Este trabalho foi realizado no âmbito do Centro de Inteligência Artificial da Universidade de São Paulo (C4AI - http://c4ai.inova.usp.br/), com o apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (processo FAPESP #2019/07665-4) e da IBM. Este projeto também foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações, com recursos da Lei N. 8.248, de 23 de outubro de 1991, no âmbito do PPI-Softex, coordenado pela Softex e publicado como Residência em TIC 13, DOU 01245.010222/2022-44.

Referências

Afonso, S., Bick, E., Haber, R., e Santos, D. (2002). Floresta sinta(c)tica: A treebank for portuguese. In the *Proceedings of the Conference on Language Resources and Evaluation* (LREC), 1698–1703.

Baker, J. K. (1979). Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 132–132.

Bannard, C., Lieven, E., e Tomasello, M. (2009). Modeling children's early grammatical knowledge. In the *Proceedings of the National Academy of Sciences* (PNAS), 17284–17289.

Bengio, Y., Louradour, J., Collobert, R., e Weston, J. (2009). Curriculum learning. In the *Proceedings of the 26th Annual International Conference on Machine Learning* (ICML), 41–48.

Blei, D. M. e Lafferty, J. D. (2005). Correlated topic models. In the *Proceedings of Advances in Neural Information Processing Systems* (NIPS), 147–154.

Bod, R. (2007). Is the end of supervised parsing in sight? In the *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics* (ACL), 400–407.

Bresnan, J., Asudeh, A., Toivonen, I., e Wechsler, S. (2015). Lexical-functional syntax. John Wiley & Sons.

Chomsky, N. (2014). Aspects of the Theory of Syntax, volume 11. MIT press.

Cohen, S. B. e Smith, N. A. (2009). Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In the *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics* (NAACL), 74–82.

da Costa, P. B. e Kepler, F. N. (2014). Semi-supervised parsing of portuguese. In the *Proceedings of the Computational Processing of the Portuguese Language - 11th International Conference* (PROPOR), 102–107.
Dahl, V., Bel-Enguix, G., Tirado, V., e Miralles, J. E. (2023). Grammar induction for under-resourced languages: The case of ch'ol. In the *Proceedings of the Analysis, Verification and Transformation for Declarative Programming and Intelligent Systems - Essays Dedicated to Manuel Hermenegildo on the Occasion of His 60th Birthday*, 113–132.
de Marneffe, M.-C., Manning, C. D., Nivre, J., e Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, 255–308.
de Souza, E., Silveira, A., Cavalcanti, T., Castro, M. C., e Freitas, C. (2021). Petrogold corpus padrão ouro para o domínio do petroleo. *In Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana* (STIL), 29–38.
Drozdov, A., Verga, P., Yadav, M., Iyyer, M., e McCallum, A. (2019). Unsupervised latent tree induction with deep inside-outside recursive autoencoders. In the *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics* (NAACL), 1129-1141.
Klein, D. e Manning, C. D. (2002). A generative constituent-context model for improved grammar induction. In the *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL), 128–135.
Futrell, R., Qian, P., Gibson, E., Fedorenko, E., e Blank, I. (2019). Syntactic dependencies correspond to word pairs with high mutual information. In the *Proceedings of the fifth international conference on dependency linguistics* (depling), 3–13.
Han, W., Jiang, Y., e Tu, K. (2017). Dependency grammar induction with neural lexicalization and big training data. In the *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (EMNLP), 1683–1688.
Han, W., Jiang, Y., e Tu, K. (2019a). Enhancing unsupervised generative dependency parser with contextual information. In the *Proceedings of the 57th Conference of the Association for Computational Linguistics* (ACL), 5315–5325.
Han, W., Jiang, Y., e Tu, K. (2019b). Lexicalized neural unsupervised dependency parsing. *Neurocomputing*, 105–115.
Hartmann, N., Fonseca, E. R., Shulby, C., Treviso, M. V., Rodrigues, J. S., e Alu´ısio, S. M. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In the *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology* (STIL), 122–131.
Hoover, J. L., Du, W., Sordoni, A., e O'Donnell, T. J. (2021). Linguistic dependencies and statistical dependence. In the *Proceedings of Conference on Empirical Methods in Natural Language Processing* (EMNLP), 2941–2963.
Headden III, W. P., Johnson, M., e McClosky, D. (2009). Improving unsupervised dependency parsing with richer contexts and smoothing. In the *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics* (NAACL), 101–109.
Jiang, Y., Han, W., e Tu, K. (2016). Unsupervised neural dependency parsing. In the *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (EMNLP), 763–771.
Klein, D. e Manning, C. D. (2004). Corpus-based induction of syntactic structure:

Models of dependency and constituency. In the *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (ACL), 478–485.
Lin, B., Yao, Z., Shi, J., Cao, S., Tang, B., Li, S., Luo, Y., Li, J., e Hou, L. (2022). Dependency parsing via sequence generation. *Findings of the Association for Computational Linguistics*, 7339–7353.
Linguateca (2023). Cetem publico: Um corpus de grandes dimensões de linguagem jornalística portuguesa. Linguateca,http://www.linguateca.pt/CETENFolha/, última visita: Junho de 2023.
Magerman, D. M. e Marcus, M.a P. (1990). Parsing a natural language using mutual information statistics. In the *Proceedings of the 8th National Conference on Artificial Intelligence* (AAAI), 984–989.
Pardo, T. A. S., Duran, M. S., Lopes, L., Felippo, A. d., Roman, N. T., e Nunes, M. d. G. V. (2021). Porttinari: a large multi-genre treebank for brazilian portuguese. In the *Proceedings of the XIII Symposium in Information and Human Language* (STIL), 1–10.
Pate, J. K. e Johnson, M. (2016). Grammar induction from (lots of) words alone. In the *Proceedings of 26th International Conference on Computational Linguistics* (COLING), 23–32.
Seginer, Y. (2007). Fast unsupervised incremental parsing. In the *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics* (ACL), 384–391
Shen, Y., Tay, Y., Zheng, C., Bahri, D., Metzler, D., e Courville, A. C. (2021). Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In the *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (ACL/IJNLP), 7196–7209.
Spitkovsky, V. I., Alshawi, H., e Jurafsky, D. (2010). From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In the *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics* (NAACL), 751–759.
Spitkovsky, V. I., Alshawi, H., e Jurafsky, D. (2013). Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In the *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (EMNLP), 1983–1995.
Stevenson, A. e Cordy, J. R. (2014). A survey of grammatical inference in software engineering. *Science of Computer Programming*, 444–459.
Theodor, C. C. e Siebert-Cole, E. (2020). Family tree of languages. https://www.researchgate.net/publication/342850691 TREES of LANGUAGES 2022, ultima visita:junho 2023.
Unold, O., Gabor, M., e Dyrka, W. (2020). Unsupervised grammar induction for revealing the internal structure of protein sequence motifs. In the *Proceedings of Artificial Intelligence in Medicine: 18th International Conference on Artificial Intelligence in Medicine* (AIME), 299–309.
Yang, S., Jiang, Y., Han, W., e Tu, K. (2020). Second-order unsupervised neural dependency parsing. In the *Proceedings of the 28th International Conference on Computational Linguistics* (COLING), 3911–3924

CAPÍTULO

5

GRAMMAR INDUCTION FOR BRAZILIAN INDIGENOUS LANGUAGES

O trabalho intitulado ´´*Grammar induction for brazilian indigenous languages"* tem como os autores *Diego Pedro Gonçalves da Silva* e *Thiago Alexandre Salgueiro Pardo*, e foi publicado no workshop *"NLP for indigenous langauges of Lusophone Countries"*, evento satélite do *International Conference on Computational Processing of Portuguese – PROPOR*. Este trabalho foi motivado pelo fato de a maioria das línguas indígenas brasileiras não serem digitalizadas. Atualmente há um número muito pequeno de córpus em línguas indígenas. Essa situação dificulta a criação de ferramentas computacionais voltada para as línguas indígenas. Isso motivou a implementação de um modelo simples que não requer uma grande quantidade de dados para ser treinada com o objetivo de induzir relações sintáticas das línguas indígenas. As contribuições relevantes para essa dissertação foi em demonstrar que é possível induzir gramática em uma pequena quantidade de dados utilizando um método simples baseado na informação mútua.

Using Mutual Information to discover dependency relations across 69 languages

Anonymous submission

Abstract

Discovering language structure is very important for several Natural Language Processing applications, which has fostered investigations on automatic grammar induction. In such a research topic, a probabilistic measure – Mutual Information – has been used with some success for identifying word pairs that are more likely to be in some syntactical relationship. Despite the use of mutual information being old and well-known in Natural Language Processing, there is a lack of research on the universal syntactic relation identification in languages of different typologies. This paper presents an in-depth study of Mutual Information for a significant number of languages, following the Universal Dependencies framework. We run experiments for 69 languages, but give special emphasis on four of them that belong to different families: Chinese, English, Arabic, and Basque to testify mutual information utility. Overall, the results indicate that Mutual Information may be used to discover dependency relationships within a distance limit between the words in the sentences.

Keywords: Grammar Induction, Mutual Information, Universal Dependencies

1. Introduction

Understanding the structure of the language and how it works is fundamental to several knowledge domains. In the research field of Natural Language Processing (NLP), in particular, it may be useful for producing robust and high-quality applications, such as machine translation (which needs to know syntax to produce the appropriate word order in the target language) and text simplification (which may rewrite more complex syntactic structures, as subordinate and coordinate clauses, in order to produce shorter and easier-to-read sentences), whether the language structure is explicitly modeled or is indirectly learned by machine learning strategies (as in recent deep learning approaches, e.g., the transformer architecture (Vaswani et al., 2017)).

Distinct linguistic views influence many methods used in NLP. There are symbolic approaches, often grounded in generative grammar, and probabilistic methods, including those based on the distributional hypothesis (Harris, 1954). This hypothesis suggests that words appearing in similar contexts tend to have similar meanings, though capturing this relationship explicitly can be challenging. Well-known metrics like Mutual Information (MI) and modern language modeling techniques utilize concepts from the distributional hypothesis.

MI is of special interest in this paper. It is a metric that is relatively simple to understand and has been used by NLP researchers for several different purposes. It presents good results applied to dependency grammar (Solan et al., 2005) and has been widely used to investigate language structuring (Church and Hanks, 1990). The majority of studies about language structures that use MI have been focused on specific languages, such as Japanese (de Paiva Alves, 1996), Portuguese (da Silva and Pardo, 2023), and English (Futrell et al., 2019; Hoover et al., 2021), mainly due to a lack of treebanks for different languages. However, the Universal Dependencies framework (UD) (de Marneffe et al., 2021) has changed this scenario, providing several datasets in dependency grammar for different languages, and contributing to a more exhaustive and intricate investigation.

Dependency grammar offers an alternative to constituency grammar for representing grammatical structures. This formalism demonstrates how words in a sentence relate to one another, functioning as either dependents or heads within their relationships, as described by Hays (1964). In this formalism, a sentence can be generated by rules of the form

$$\begin{aligned} I - *(X) \\ II - X(X_1 ... X_i * X_{i+1} .. X_n) \end{aligned} \quad (1)$$

The second rule is a decomposition from the first rule. The star * represents the parent term, which is the term outside the brackets, in the tree. In the case of the first rule, the parent is assigned to *root*. For instance, in the sentence "The cat chased the dog", it is possible to establish a subject relationship between the verb "chased", which would be the head of the relation, and the noun "cat", the dependent[1]. That sentence can be generated by using the first rule described in 1

$$*(V)$$

After the second rule applied in V

$$*(V(N, *, N))$$

After the third rule choice

$$*(V(N(D, *), *, N(*, D)))$$

[1] For the interested reader, a constituency grammar would claim that "the cat" is a noun phrase that, with the following verb phrase, form the sentence.

This paper presents a comprehensive investigation of MI across 69 distinct languages obtained from the UD initiative. Our study aims to assess the viability of employing MI to uncover dependency structures in every language. We highlight the results in four different languages from different families to support that hypothesis: English (Indo-European) (Bouckaert et al., 2012), Basque (which forms a language family on its own) (Izagirre and Alonso, 2021), Arabic (Afro-Asiatic) (Gragg, 2019) and Chinese (Sino-Tibetan) (Thurgood and LaPolla, 2016). Interestingly, we also uncover which syntactic relations are easier to detect, despite the typology differences. Moreover, we present results for West Iberian languages (Portuguese, Catalan, Spanish, and Galician).

Such investigation is relevant for NLP practical purposes, as it may indicate that measures such as MI may aid in automatic grammar induction and parsing techniques, as demonstrated by Futrell et al. (2015), for detecting words that are likely to show syntactical relationships or helping to identify parsing errors. With a more linguistic view, it may be useful to discover language structure for severe low-resource languages (da Silva and Pardo, 2024) or even dead languages.

In what follows, we briefly overview the main related work in the area. Section 3 brings the adopted methodology. Section 4 describes the achieved results. Section 5 presents a discussion about this work and 6 brings some final remarks.

2. Related work

MI has evolved through various definitions (Zeng et al., 2015); we adopted the modern definition used in Information Theory described in Equation 2[2]. MI is computed by the joint distribution of two variables considered to be dependent on each other – $P(x, y)$, where y is a word that appears after the word x – divided by the joint distribution of the same variables considered to be independent – $P(x)P(y)$. If the two variables are completely independent (e.g., they never appear together, such as in the sentence "Hydrogen cow"), then the joint probability will be equal to zero. However, if the two variables are completely dependent, MI will be higher than one, given that the joint probability will always be higher than the product of the probabilities. As an example, consider the tokens *"the"* and *"in"* in English, which are very common, with $P(the) = 0.0484$ and $P(in) = 0.0167$ (as computed using an English corpus): the MI between these words is negative because $P(the, in) = 0.00006$ and $P(the)P(in) = 0.0008$, resulting in $log\frac{P(the,in)}{P(the)(in)} < 0$ (it is usual to substitute all *negative* MI by zero). On the other hand, *"the"* and *"of"* are also very common in English and common to occur together, with $P(the) = 0.0484$, $P(of) = 0.0211$ and $P(the, of) = 0.005$: MI would be positive, with $P(the, of)log_2\frac{P(the,of)}{P(the)(of)} = 0.011$.

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) log_2 \frac{P(x, y)}{P(x)P(y)} \quad (2)$$

MI has been studied in many contexts of NLP, including parsing (de Paiva Alves, 1996), grammar induction (Solan et al., 2005), and analyzing linguistic structures (Futrell et al., 2019; Hoover et al., 2021; da Silva and Pardo, 2023).

de Paiva Alves (1996) employed MI to select the most likely dependency structure in Japanese. Each likely relation from the dependency structure was previously generated based on part-of-speech and syntactic features. To select the most probable relation, the author used a dictionary taxonomy with more than 1 million co-occurrences between words to compute MI. Solan et al. (2005) pursued the unsupervised induction of constituency grammar in English by utilizing the ADIOS framework (Solan et al., 2002) to segment text into the smallest morphological constituents for MI computation.

Futrell et al. (2019) discovered that MI may be used to distinguish syntactically related word pairs from word pairs without dependency relations. Their study used distribution information over 320 million tokens. Hoover et al. (2021) extends Futrell et al. (2019), integrating pre-trained language models. Further extending this approach, da Silva and Pardo (2023) examined syntactic relations in Portuguese, demonstrating variations in MI scores across different syntactic relations.

It is important to say that, despite following different research lines, some works have already used a large number of languages to identify universals. For instance, Futrell et al. (2015) analyzed dependency length in 37 languages, while Malik-Moraleda et al. (2022) investigated universal networks in 45 languages. Moreover, Yu et al. (2018) examined Zipf's law patterns in 50 languages.

Next section presents the methodology that we adopt in this paper, including the datasets and the experiment setup.

3. Methodology

We use data from UD version 2.12 [3]. It contains 245 treebanks for 141 languages. Given the heterogeneous composition of treebanks within the

[2]The reader should notice that this equation is different from pointwise mutual information, which aims to measure the correlation between two variables. MI is the expectation of pointwise mutual information over all possible outcomes.

[3]https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-5150

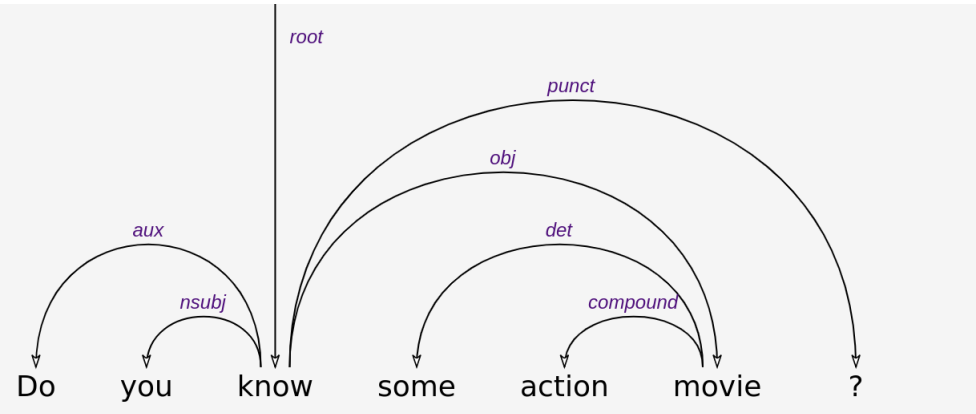


Figure 1: Example of annotated sentence for English

UD repository, with some containing solely training data and others predominantly comprising test data, we exclusively employ treebanks that include both test and training sets. Moreover, we require that the training dataset represent at least 50% of the overall sentences available in the treebank. About 131 treebanks for 69 distinct languages meet these criteria. Overall, UD treebanks comprise 1.3 million sentences and 21.1 million tokens in the training datasets. The 131 treebanks in this study consist of 162K sentences and 2.7 million tokens in the training datasets. The distribution of sentence lengths in the training dataset conforms to a normal distribution, specifically denoted as $\sim \mathcal{N}(17.66, 6.67)$. Approximately 14.4% of all tokens analyzed in our study pertains to the German language, with the Czech language accounting for 8.5%, and Russian representing 6.8%.

We follow the UD framework to consider variations within a single language as distinct languages such as *French* and *Old French*. Thus, these languages are treated differently in this study. The treebanks for each language within the same dataset are combined, with a division between test and training data. For instance, the English language has ten treebanks, and eight of then meet the selection criteria. These eight datasets are combined into one, discriminating test and training. All text is converted to lowercase, and punctuation is kept. The root dependency relation is also included in the experiments.

The main idea behind the methodology utilized in this study involves pairing two words in all possible configurations, calculated as $\binom{n}{2} = \frac{n!}{2(n-2)!}$, where n represents the length of the sentence. Subsequently, the mutual information of these word pairs is computed, and the n pairs exhibiting the highest mutual information are selected as those probably presenting syntactic relations. Then, these supposedly related word pairs are compared to manually annotated ones, determining the performance of the approach.

In the experiments involving all languages, we performed four different settings based on the distance between the words (DW) in the sentence (1, 2, 5, and 9 words). The sentence length N was fixed at ten tokens (including punctuation) to reduce computational processing, resulting in a total of eight different parameter configurations. We chose these values based on the data distribution. In the training dataset, approximately 42.8% of all dependency relations have a distance between the terms equal to one, while 63.5% demonstrate a maximum distance of two. Notably, a distance of five or less includes 84% of all relations, with a distance at most nine accounting for 93%. In contrast, only 36% of all sentences have a size of no more than ten tokens, resulting in the exclusion of 64% of sentences from this study. For languages from different families, we use only sentences and a distance equal to nine as configuration. We chose a different configuration to isolate these variables to analyze better the syntactic relation used in the language. For both experiments, we divided into two groups: using all part-of-speech tags and using only part-of-speech tags that represent open class words according to the Universal Dependencies framework (adjective, adverb, interjection, noun, proper noun, and verb) to try to deal with the most important relations (as nsubj, obj and iobj). In an open class words setting, a dependency relation that does not have both words within the open class words set is discarded from calculations.

Employing the test set, in the experiments involving all languages, we discarded all sentences with less than three tokens because small sentences may have a small number of influential syntactic relations. We conducted permutations of pairs of words within each sentence based on each configuration. We combine the sentence using binomial coefficients $\binom{n}{k}$, where *k* is two (tokens per pair) and *n* is the number of tokens present in the sen-

tence, including punctuation. The total number of permuted pairs is described by $\sum_{d=1}^{DW} n - d$, where n is the number of tokens in the sentence, including punctuation.

This permutation process formed the final set of the Sentence Permutation (SP), consisting of pairs of tokens, where the first token precedes the second in the sentence's sequence. Subsequently, we calculated the MI for each word pair in the SP. As our primary interest lies in evaluating the significance of MI for discerning dependency relations, and for sparsity reasons, we confined this experiment to the top ten pairs exhibiting the highest MI. Some permutation pairs may have zero MI; in this case, the ten highest MI are sorted lexicographically.

To evaluate our experiments, we rely on two metrics: Intersection Set Accuracy (ISC) and Coverage Accuracy (CA). The first one computes the dependency relation match ratio by the number of SP, while the second one computes the dependency relation match ratio by the number of Annotated Dependency Relations set (ADR). These two metrics were proposed instead of just one because the size of sentences varies, while the number of SP is fixed along the same experiment. Both metrics yield scores within the interval *[0,1]*. In our studies, the SP will always be less or equal to the length of ADR. Given the set of all sentences S and the number of pairs of each sentence to be evaluated from 1 to p, which p is the length of the sentence, Equation 3 computes the ISC score for each treebank.

$$ISC_{S,p} = \frac{\sum_{s \in S} SP_s\{1..p\} \cap ADR_s}{\sum_{s \in S} \sum_{i=1}^{p} i} \quad (3)$$

Equation 4 characterizes the second metric, CA. Unlike ISC, which aims to compute the accuracy of the number of pairs, CA aims to compute accuracy by the amount of dependency relations. Sentences with over ten relations may exhibit lower performance as we focus on the top 10 pairs.

$$CA_{S,p} = \frac{\sum_{s \in S} SP_{1..p} \cap ADR_s}{\sum_{s \in S} \sum ADR_s} \quad (4)$$

In our study, we do not consider the direction of the dependency relation, which could be an object of future investigations.

To aid the understanding of the CA and ISC equations, consider the sentence in Figure 1 as an example. Using ten pairs in permutations without constraints about the distance between tokens, we have $\binom{7}{2} = 21$ combinations. We describe the following 21 combination pairs as a tuple of three terms: the first is the actual token, the second is the successor token, and the last one is the MI score for both words. The following sequence of pairs follows the order in which they were formed (token$_1$|token$_2$ and token$_2$|token$_3$, for example): (*do*|*you*|0.0016), (*do*|*know*|1.87×10^{-5}), (*do*|*some*|2.66×10^{-5}), (*do*|*action*|0),(*do*|*move*|6.969×10^{-6}), (*do*|?|6.46×10^{-5}), (*you*|*know*|0.0020) ,(*you*|*some*|7.92×10^{-6}), (*you*|*action*|0), (*you*|*movie*|0), (*you*|?|5.78×10^{-5}), (*know*|*some*|3.08×10^{-6}), (*know*|*action*|0), (*know*|*movie*|0), (*know*|?|5.87×10^{-5}), (*some*|*action*|0), (*some*|*movie*|0), (*some*|?|0), (*action*|*movie*|0), (*action*|?|0), (*movie*|?|7.01×10^{-5}).

From these pairs, the ten pairs with the highest MI are $SP_s = \{$*(do|you)*, *(you|know)*, *(do|you)*, *(know|?)*, *(you|?)*, *(do|some)*, *(do|know)*, *(you|some)*, *(movie|?)*, *(know|some)*$\}$. These pairs match three dependency relations in the sentence presented in Figure 1: *(know|?)*, *(do|know)*, *(you|know)*. That set is the result of $SP_{Figure1} \cap ADR_{Figure1}$. Now, to compute ISC, we divide four by the number of pairs, resulting in a score of 0.3. To compute the CA score, we divide four by the number of dependency relations, which is 7, resulting in 0.42.

4. Results

In Section 4.1, we present the results across all languages, and in Section 4.2, we focus on four languages from different families.

4.1. General results

The scores presented in Tables 1, 2, 3, and 4 are referent to the percentage of the respective score for all languages combined. They show how many syntactic relations can be discovered by using the metrics ISC and CA attending DW criteria for each cumulative number of pairs. The first column informs the permutation constraints in the sentence. For instance, in Figure 1, the syntactic relation <do, know> has $DW = 2$, while the syntactic relation <know, .> has $DW = 4$. The second column shows the number of syntactic relations present in all languages evaluated. The third to the twelfth columns present the cumulative percentage for each pair. In pair 1, the $DRA_s \cap SP_s\{1\}$ considers only the pair with the highest MI. In pair 2, $DRA_s \cap SP_s\{1..2\}$, the two pairs with the highest MI in SP_s are used to intersect with the set DRA_s.

Tables 1 and 2 show ISC scores across all languages for all word classes and open class words, respectively. In both tables, for all word distances, the ISC score decreases as the number of pairs increases. There are two factors contributing to this. Firstly, the data tends to be sparse, leading to a higher likelihood of zero mutual information as the number of pairs increases. Second, with more distant pairs, the mutual information decreases, making it more challenging to identify dependency relations accurately.

Table 1: ISC all word classes (percentage)

DW	dep relations	Pairs									
		1	2	3	4	5	6	7	8	9	10
1	3630920	*53.4*	*54.9*	*54.6*	*54.2*	*53.8*	*53.4*	*53.1*	*52.8*	*52.6*	*52.6*
2	3630920	*49.5*	*49.8*	*48.8*	*47.7*	*46.6*	*45.8*	*45.0*	*44.5*	*43.8*	*43.4*
5	3630920	*45.6*	*45.2*	*43.9*	*42.4*	*41.1*	*39.8*	*38.9*	*38.1*	*37.3*	*36.4*
9	3630920	*44.6*	*44.3*	*42.9*	*41.5*	*40.3*	*39.0*	*38.1*	*37.3*	*36.5*	*35.6*

Another crucial result in the tables is the difference in scores between DW. That is to be expected, as the increase in the number of pairs combinations contributes to a higher likelihood of encountering noise from frequent tokens, which can make it more challenging to identify dependency relations correctly. For example, for a sentence of ten tokens when DW is set to 2, there are only 17 pairs, whereas with DW set to 9, there are 45 pair combinations. Languages such as English, which use fewer closed class has a good performance in extracting 'nsubj' and 'obj' relations. However, in other languages that have more closed class words, such as Portuguese, it becomes more challenging to extract these influential dependency relations due to the prevalence of closed class words. Figure 2 illustrates det and case marking (case), the syntactic relation in which the tokens described before are more frequent, following the Zipfian law. In Table 2, although it still occurs, the scores are higher because of the restriction on the part-of-speech categories considered in the combination. By exclusively utilizing open class words, the ISC score exhibited a 57% increase on pair 1 in DW 9 compared to the ISC score derived from all word classes. The disparities in ISC scores between Tables 1 and 2 diminish as DW increases, except when using pair one due to the lower noise.

Tables 3 and 4 show CA scores across all languages for open and all classes. Unlike ISC scores described in 1 and 2, the CA score increases as the number of pairs increases. It is natural to occur since the coverage increases as the number of pairs increases, but the rate decreases over pairs for the same reason as the ISC score: lower mutual information. CA score decreases as DW increases for the same reason as the ISC score. In Table 4, the difference is higher in ISC score between the two groups (all word classes and only open class words). In DW 9 on pair one, the score is 165% higher than using all word classes. It happens because, given that the space is reduced, it is easier to retrieve the correct dependency relations. However, differently from ISC scores, as the number of pairs increases, the difference between the scores in Tables 3 and 4 decreases , even with lower scores highlighted in bold. The main contribution to using two different scores in Tables 1, 2, 3, and 4 is the way to see the scores. While ISC normalizes the score by the number of pairs, CA normalizes by the number of dependency relations. It allows to see the results in different perspectives.

Relation	1	2	3	4	5	6	7	8	9	10	Total
advcl	2.2	7.3	11.6	15.7	19.8	23.9	27.3	30.7	33.9	37.0	19,364
advmod	8.5	18.2	26.5	34.0	40.9	46.7	51.6	55.8	59.3	62.2	103,196
amod	9.9	19.6	28.8	37.3	45.1	51.8	57.8	62.8	66.8	69.7	59,356
aux	10.0	22.7	32.1	39.3	45.7	51.1	55.6	59.4	62.4	64.9	28,148
case	16.9	29.6	39.6	47.2	53.6	59.0	63.3	66.8	69.7	72.1	88,608
ccomp	1.5	5.2	11.2	14.8	18.8	22.7	26.0	29.9	33.1	35.9	10,964
cop	6.6	15.4	24.3	32.1	39.3	45.5	50.6	55.2	58.8	61.9	24,040
csubj	2.0	7.1	12.2	17.1	21.3	25.3	29.5	33.5	36.5	39.6	3,628
det	14.6	30.0	42.1	51.4	58.7	64.4	68.9	72.7	75.7	78.0	60,744
iobj	8.3	17.0	25.0	32.5	39.0	44.4	49.2	53.3	56.8	59.8	7,024
nmod	6.1	13.2	20.7	27.3	33.4	39.0	43.8	48.3	52.1	55.2	81,600
nsubj	4.9	12.5	20.2	27.0	33.3	38.6	43.3	47.3	50.9	53.9	133,252
obj	6.8	15.5	24.1	31.8	38.7	44.7	49.7	53.9	57.6	60.7	85,656
obl	2.8	6.6	11.3	16.0	21.0	25.8	30.0	34.1	37.8	41.1	89,100
xcomp	4.7	10.5	17.2	23.8	30.0	35.3	40.4	45.0	48.9	51.8	15,928

Figure 2: CA all word classes (percentage)

We also wanted to analyze the performance of detecting specific syntactic relations. For this experiment, we apply DW equal to 1 using the all word classes setting. We constrained the experiment to sentences with a maximum length of ten tokens. Figure 2 provides a comprehensive overview of the CA scores concerning a relative CA score over all occurrences of the syntactic relation across all languages. We filtered out syntactic relations accounting for less than 1% of occurrences. The scores were then weighted and averaged based on the size of each language dataset. The first column presents the syntactic relations. Columns two through eleven detail the percentage of how many syntactic relation can be discovered using mutual information. For instance, using one pair, only 4.9% of all nsubj syntactic relations can be retrieved to match the dependency relations in sentences with a maximum of ten tokens. However, with ten pairs, MI can retrieve 53.9% of nsubj syntactic relations. The last columns display the total occurrences of syntactic relations in the experiment. Some syntactic relations have advantages, such as det and case, due to the average distance between terms in the relations.

It is important to emphasize that these findings do not delineate the preeminent syntactic relation, as they are contingent upon the correlation with Zipf's law and the sparsity of data. For example,

Table 2: ISC open class words (percentage)

DW	dep relations	Pairs									
		1	2	3	4	5	6	7	8	9	10
1	1792140	*80.7*	*76.0*	*73.7*	*72.6*	*72.1*	*71.9*	*71.8*	*71.8*	*71.8*	*71.8*
2	1953040	*76.7*	*69.3*	*65.9*	*63.5*	*61.9*	*60.9*	*60.2*	*59.8*	*59.5*	*59.3*
5	1995350	*71.0*	*61.5*	*57.5*	*54.7*	*52.2*	*50.7*	*49.6*	*48.5*	*47.7*	*47.1*
9	1996640	*70.2*	*60.5*	*56.3*	*53.5*	*50.9*	*49.3*	*48.1*	*47.1*	*46.0*	*45.3*

Table 3: CA all classes (percentage)

DW	dep relations	Pairs									
		1	2	3	4	5	6	7	8	9	10
1	3630920	*8.2*	*16.8*	*24.2*	*30.4*	*35.4*	*39.3*	*42.0*	*43.8*	*44.6*	*44.6*
2	3630920	*7.6*	*15.2*	*22.4*	*28.4*	*34.2*	*39.0*	*43.6*	*47.4*	*50.9*	*53.8*
5	3630920	*7.0*	*13.9*	*20.1*	*25.3*	*30.2*	*34.7*	*38.5*	*42.2*	*45.7*	*48.9*
9	3630920	*6.9*	*13.6*	*19.7*	*24.8*	*29.6*	*34.0*	*37.7*	*41.3*	*44.7*	*47.8*

the ten most prevalent tokens in nsubj and obj, which primarily refer to the subject and object of a verb in, jointly constitute 14.7% and 10.6% of the total occurrences, respectively. In contrast, det and case account for 33.9% and 40.9% of their respective syntactic relations.

Additionally, we investigated the application of MI, focusing on different languages. The outcomes for all languages in both setting configurations concerning word classes are presented in Table 5. These findings show that the score exhibits considerable variation with $\sigma = 3.25$ for the integral data but is even higher when limited to open word classes with $\sigma = 5.58$. To limit the experiment to open word classes improves the ISC for all languages, except for Korean, which slightly decreased in 3%. The average improvement using only open word classes was 40.2%. An interesting finding refers to the writing system. Languages derived from the Greek alphabet or Cyrillic script present an improvement of 44.5%. In contrast, languages from different writing systems showed an improvement of only 24% using open word classes.

Moreover, we did not find any indication of an association between the treebank size and the language's performance (using all word classes) using ISC score, as well as vocabulary size. However, a negative correlation of -0.40 was identified with the proportion of sentences containing duplicate tokens, as determined by the Spearman correlation. For instance, in the sentence *"Choose a **country**, any **country**"*, the token *"country"* is repeated. Approximately 45.4% of the sentences within the treebanks present duplicate tokens. In specific languages, such as ancient Greek and Galician, this proportion exceeds 90%. Conversely, in languages with writing systems unrelated to the Greek alphabetic or Cyrillic script, like Telugu and Turkish, it is below 20%. Despite this prevalent repetition, the presence of duplicate dependency relations, when constrained to identical tokens, remains under 5%, and, when considering both exact tokens, syntactic relations, and parts-of-speech, it registers less than 0.1% of the sentences within the treebanks.

Given that 5% of the sentences contain more tokens than unique types, it is inevitable that specific pairs may remain unmatchable due to duplication, resulting in a decrease in the overall ISC score. Conversely, we did not observe any significant correlation between the CA score and factors such as treebank size, vocabulary size, or duplicate tokens. It is important to highlight that dependency relations that has the same part-of-speech, token form, and syntactic relations are treated as a single relation in the sentence. Instances where an identical dependency relation occurs result in the systematic exclusion of all corresponding occurrences from scoring calculations. For example, in the sentence "I bought an apple cake and an apple candy" the relation ‹an, apple› is duplicated, therefore the model computes it only once.

According to (Nivre et al., 2020), not all syntactic relations defined in the UD framework can be considered strictly a dependency relation. However, certain syntactic relations exhibit a higher degree of dependency than others. For this reason, we exclusively consider the core arguments of clausal predicates, non-core dependents of clausal predicates, and dependents of nominal structures, which are dependency relations in a narrow sense (Nivre et al., 2020). Core arguments primarily consist of the subject (nsubj) and object (obj). Non-core dependents are composed mainly of oblique nominal (obl), adverbial modifier (advmod), and adverbial clause modifier (advcl). Lastly, dependents of nominals are predominantly characterized by case, nominal modifiers (nmod), and det.

Table 4: CA open class words (percentage)

DW	dep relations	Pairs									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	1792140	*19.5*	*30.3*	*35.7*	*38.2*	*39.4*	*39.9*	*40.0*	*40.1*	*40.1*	*40.1*
2	1953040	*19.5*	*31.3*	*40.0*	*45.2*	*49.1*	*51.3*	*53.0*	*53.9*	*54.7*	*55.0*
5	1995350	*18.5*	*29.0*	*39.1*	*45.1*	*50.5*	*55.6*	*58.4*	*61.1*	*63.5*	*65.5*
9	1996640	*18.3*	*28.6*	*38.5*	*44.3*	*49.6*	*55.1*	*57.9*	*60.6*	*63.1*	*65.9*

Table 5: ISC score on different languages

Language	**all**	**open**	**Language**	**all**	**open**
Afrikaans	33.6	55.3	Latvian	39.0	58.2
Ancient_Greek	37.8	46.6	Ligurian	34.9	62.4
Ancient_Hebrew	41.9	51.4	Lithuanian	38.8	55.0
Arabic	46.2	55.7	Maghrebi_Arabic_French	43.0	53.5
Armenian	40.3	56.6	Maltese	38.5	63.9
Basque	38.8	53.8	Marathi	41.6	57.1
Belarusian	41.3	56.0	Naija	40.2	67.3
Bulgarian	37.8	57.6	North_Sami	38.2	53.9
Catalan	39.4	63.0	Norwegian	38.4	59.2
Chinese	36.2	40.8	Old_Church_Slavonic	42.8	53.5
Classical_Chinese	48.3	54.0	Old_East_Slavic	41.7	54.1
Coptic	34.5	70.7	Old_French	39.8	61.1
Croatian	37.6	53.3	Persian	44.3	48.8
Czech	39.9	56.2	Polish	40.1	59.8
Danish	37.1	61.0	Pomak	39.0	63.9
Dutch	39.5	57.4	Portuguese	42.1	61.9
English	41.2	61.3	Romanian	40.9	62.3
Estonian	39.0	52.4	Russian	40.8	56.9
Faroese	41.7	65.8	Sanskrit	41.1	48.1
Finnish	43.0	54.4	Scottish_Gaelic	42.1	58.1
French	41.4	66.2	Serbian	37.1	52.3
Galician	39.6	62.5	Slovak	39.3	54.8
German	41.6	57.6	Slovenian	37.5	60.3
Gothic	41.8	52.3	Spanish	39.8	64.1
Greek	36.2	66.7	Swedish	37.3	58.6
Hebrew	40.9	61.2	Tamil	42.2	49.4
Hindi	41.4	56.3	Turkish	49.6	54.5
Hungarian	38.6	53.6	Telugu	50.9	61.4
Icelandic	44.0	60.0	Ukrainian	39.2	58.4
Indonesian	39.6	56.2	Urdu	40.6	54.4
Irish	38.0	56.7	Uyghur	44.6	50.1
Italian	39.1	64.2	Vietnamese	37.6	52.7
Japanese	46.0	60.0	Western_Armenian	37.3	54.3
Korean	45.8	44.4	Wolof	37.1	63.1
Latin	41.2	52.1			

4.2. Results for different language families

As we expected, there is a big difference between language families in how mutual information can be useful. For instance, in Arabic, the majority of the highest MI pairs in the sentence are modifiers (amod and nmod). However, these scores decrease when applied only to open word classes. Another crucial observation pertains to the nsubj syntactic relation, which exhibits a notable performance in Japanese and English enhancement when exclusively considering open word classes, leading to it being the top-performing class. Specifically, utilizing the ten highest pairs, the nsubj syntactic relation attained 49.3% for Japanese, 40.8% for English, 11.4% for Basque, and 6.2% for Arabic. The Japanese language demonstrated better results in open word classes. On the other hand, Arabic does not demonstrate good performance.

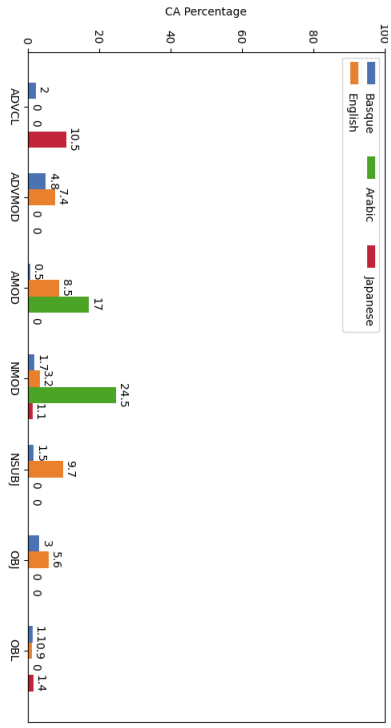


Figure 3: CA Score for all word classes

These findings demonstrated that languages from different families presents different results when using MI to identify dependency relations

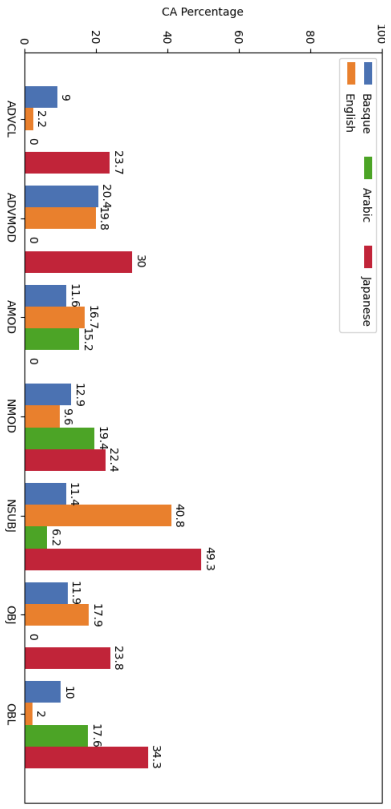


Figure 4: CA Score for only open class words

4.3. West Iberian languages

The West Iberian languages include languages spoken in Spain and in Portugal. The main languages spoken are Catalan, Galician, Spanish and Portuguese. The findings for open-class words and all lexical categories are notably similar, indicating that the localization of language use can have a significant impact. For example, French exhibited greater similarity to Spanish and Catalan than to Portuguese and Galician.

5. Discussion

Mutual information has long found application in NLP, but, according to our knowledge, no one has analyzed the utility of mutual information to the discovery of syntactic structures in so many languages. Despite the superior performance of neural networks in extracting syntactic structures from dependency grammar compared to mutual information, they encounter challenges in extracting dependency relations when there are considerable distances between the dependent term and the head term (Liu et al., 2017). Furthermore, these methods often entail high energy consumption. In this study, we show that the writing system could have a higher influence on syntactic dependency relation than its grammar. It is important to note that this study does not aim to propose a new alternative for syntactic structure extraction but rather seeks to provide a potential direction for integrating grammatical induction methods with modern approaches.

6. Final remarks

In conclusion, this study comprehensively examines MI across a diverse linguistic spectrum, including 69 languages from UD. The research included an in-depth investigation into the impact of MI on the discovery of dependency structures, revealing its potential within certain distance constraints between the words involved in the relationships. The findings highlight the potential of MI in identifying dependency relationships. As a potential line of investigation, we aim to conduct more experiments employing advanced smoothing techniques and expanding our dataset by incorporating a broader range of treebanks and languages in future research.

References

Remco Bouckaert, Philippe Lemey, Michael Dunn, Simon J Greenhill, Alexander V Alekseyenko, Alexei J Drummond, Russell D Gray, Marc A Suchard, and Quentin D Atkinson. 2012. Mapping the origins and expansion of

the indo-european language family. *Science*, 337(6097):957–960.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Diego Pedro Gonçalves da Silva and Thiago Alexandre Salgueiro Pardo. 2023. Induçao gramatical para o português: a contribuiçao da informaçao mutua para descoberta de relaçoes de dependência. In *Proceedings of the 14th Brazilian Symposium on Information Technology and Human Language*, pages 298–307.

Diego Pedro Gonçalves da Silva and Thiago Alexandre Salgueiro Pardo. 2024. Grammar induction for brazilian indigenous languages. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese-Vol. 2*, pages 64–72.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Eduardo de Paiva Alves. 1996. The selection of the most probable dependency structure in japanese using mutual information. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 372–374.

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Richard Futrell, Peng Qian, Edward Gibson, Evelina Fedorenko, and Idan Blank. 2019. Syntactic dependencies correspond to word pairs with high mutual information. In *Proceedings of the 5th international conference on dependency linguistics*, pages 3–13.

Gene Gragg. 2019. Semitic and afro-asiatic. *The Semitic Languages*, pages 22–48.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

David G Hays. 1964. Dependency theory: A formalism and some observations. *Language*, 40(4):511–525.

Jacob Louis Hoover, Wenyu Du, Alessandro Sordoni, and Timothy J. O'Donnell. 2021. Linguistic dependencies and statistical dependence. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2941–2963.

Neskuts Izagirre and Santos Alonso. 2021. Evolution: On the origin of basques. *Current Biology*, 31(10):R489–R490.

Haitao Liu, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews*, 21:171–193.

Saima Malik-Moraleda, Dima Ayyash, Jeanne Gallée, Josef Affourtit, Malte Hoffmann, Zachary Mineroff, Olessia Jouravlev, and Evelina Fedorenko. 2022. An investigation across 45 languages and 12 language families reveals a universal language network. *Nature Neuroscience*, 25(8):1014–1019.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4033.

Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. 2005. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, 102(33):11629–11634.

Zach Solan, Eytan Ruppin, David Horn, and Shimon Edelman. 2002. Automatic acquisition and efficient representation of syntactic structures. In *Advances in Neural Information Processing Systems*, pages 91–98.

Graham Thurgood and Randy J LaPolla. 2016. *The sino-tibetan languages*. Taylor & Francis.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Shuiyuan Yu, Chunshan Xu, and Haitao Liu. 2018. Zipf's law in 50 languages: its structural pattern, linguistic interpretation, and cognitive motivation. *arXiv preprint arXiv:1807.01855*.

Guoping Zeng et al. 2015. A unified definition of mutual information with applications in machine learning. *Mathematical Problems in Engineering*, 2015.

CAPÍTULO

6

USING MUTUAL INFORMATION TO DISCOVER DEPENDENCY RELATIONS ACROSS 69 LANGUAGES

Este estudo tem como autores *Diego Pedro Gonçalves da Silva* e *Thiago Alexandre Salgueiro Pardo* e encontra-se em fase final para submissão. Este trabalho foi motivado a partir das seguintes questões de pesquisa: A medida de Informação Mútua tem eficiência na descoberta de relações sintáticas em diferentes línguas, mesmo línguas com sistemas de escrita e sintaxe completamente diferentes? As principais contribuições desta dissertação incluem a validação dessas questões de pesquisa, o que pode contribuir para o desenvolvimento de métodos que utilizam a informação mútua em diversas línguas, inclusive aquelas com dados limitados disponíveis.

Grammar Induction for Brazilian Indigenous Languages

Diego Pedro Gonçalves da Silva and Thiago Alexandre Salgueiro Pardo
Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo
diegopedro@usp.br, taspardo@icmc.usp.br

Abstract

This paper investigates the issue of grammar induction for Brazilian indigenous languages, mainly focusing on unsupervised methods, but also testing a large language model for the task. Grammar induction poses several challenges, particularly when applied to low-resource languages, a characteristic commonly associated with indigenous languages. The primary objective in this paper is to discover syntactically related words in sentences. In addition to the contributions to linguistic studies, as in language description and structural analysis, grammar induction may help in varied Natural Language Processing tasks, as it could help detecting parsing errors, enhancing parsing results, and revealing pertinent relations for open information extraction purposes. The findings reveal that, even with a limited corpus, it is feasible to identify syntactically related words, specially for some relations. To the best of our knowledge, this represents a pioneering attempt to undertake grammar induction for Brazilian indigenous languages.
Keywords: Grammar Induction, Universal Dependencies, Mutual Information, Indigenous Languages

1. Introduction

In the year 2001, there were 6,981 languages spoken globally, some of which linguists predict will confront the threat of extinction by the year 2100 (Harrison, 2008). One of the reasons for this decline may be associated with political and social discrimination directed toward its speakers, thereby exerting an influence on subsequent generations. This influence may manifest as parents refraining from transmitting their native languages to their offspring, driven by concerns regarding perceived limitations in future opportunities (Harrison, 2008; Cruz, 2011). The consequences of a language extinction across social, political, and cultural spheres are profound and incalculable. The cumulative wisdom amassed across generations, transmitted exclusively through oral communication, irreversibly dissipates (Harrison, 2008).

In Brazil, according to data provided by *Instituto Brasileiro de Geografia e Estatística* (IBGE), there were 244 indigenous languages documented in the country in 2010 (Morello, 2016). Predominantly, these languages belong to the Tupi family, which comprises more than 40 distinct languages (Ferraz Gerardi et al., 2023). The expansive influence exerted by the Tupi language family constitutes the most extensive diffusion globally. This facilitates mutual comprehension among languages within this linguistic group, many of which share cognates (Ferraz Gerardi et al., 2023). Among the indigenous languages prevalent in Brazil, Ticuna, spoken by 46 thousand individuals, Guarani-Caiuá, with 43 thousand speakers, and Caingangue, with 37 thousand speakers, emerge as the most widely spoken ones according to IBGE (Morello, 2016). A considerable number of Brazilian indigenous languages are spoken by fewer than 100 individuals (Cruz, 2011).

Promoting literacy among indigenous children in their native language and attempting to digitalize their language constitutes strategic initiatives to mitigate language decline (Taylor, 1985; Azevedo, 2016). However, the rise of the internet may have hastened the extinction of indigenous languages, given that the prevalence of dominant languages significantly contributes to the functional loss of indigenous languages (Kornai, 2013). The content deficit of the indigenous languages adversely affects the development of technological tools for these languages, such as translation systems. These tools would be useful for disseminating information and facilitating learning, consequently, contributing to preserving the language.

Artificial Intelligence systems emerge as a significant initiative to contribute to the advance of language technologies (Pinhanez et al., 2023; de Lima et al., 2021). Addressing this challenge involves considering alternatives, such as the use of comparable texts to build parallel corpora[1], and the use of grammar induction for learning syntactical structuring patterns and lexical clustering for detecting semantically-related terms for a (probably low-resource) language of interest. Grammar induction is the focus of this paper.

In Natural Language Processing (NLP) applications, Grammar Induction (GI) proves useful for various tasks, including grammar checking, information extraction, and text simplification, to name a few. Grammar induction can be approached in an Unsupervised way (UGI), in a Semi-Supervised way (SSGI), or in a Supervised way (SGI). SGI methods demonstrated remarkable efficacy in many works, achieving accuracy rates exceeding 95% (Lin et al., 2022) for the English language, while their unsupervised counterparts present a considerable challenge, often falling short of this benchmark.

[1] It is not rare to use the Bible for such end, as it is published in many languages.

This study focuses on unsupervised approaches to induce grammar within the context of dependency paradigm, which seeks to model the dependency relations among syntactic elements. Illustrative instances are provided in the form of a Nheengatu sentence presented in Figure 1, along with its Portuguese translation portrayed in Figure 2. These sentences were extracted from the Nheengatu CompLin treebank (Avila, 2021) identified with ID *Avila2021:0:0:647*. The arrows delineate the relationships between two tokens, wherein the arrow originates from the head term and is directed toward the dependent term.

Good methods for grammar induction include Large Language Models (LLM) (Shen et al., 2021) and neural networks (He et al., 2018) and both methods need a huge amount of data for training. Due to the limited amount of available digital data in indigenous languages, we test two different approaches to discover related words in an unsupervised way: Dependency Model with Valence (DMV) (Klein and Manning., 2004), the most influential model in grammar induction tasks; and Mutual Information (MI), a measure that has demonstrated efficacy to retrieve syntactic structures (Futrell et al., 2019; Hoover et al., 2021). Furthermore, we also evaluate an LLM for the taks.

The investigation specifically centers on twelve indigenous languages spoken in Brazil, most of which were annotated as a part of the TuLaR (Tupían Language Resources) project within the "Universal Dependencies" (UD) framework (Nivre et al., 2020). Notably, seven of these languages are affiliated with the Tupi family. To the best of our knowledge, this is the first unsupervised grammar induction study within the domain of Brazilian indigenous languages. We provide the code from this project at Github[2].

Next section brings a brief literature review on the topic of grammar induction. Section 3 presents the methods that we test, while Section 4 shows and discusses the achieved results. Discussion and final remarks are presented in Sections 5 and 6.

2. Related Work

In recent decades, Grammar Induction has been applied in different contexts and diverse applications. Varied methodologies have been employed, with the DMV (Klein and Manning., 2004) emerging as the most prevalent and widely recognized approach. This approach was the first to surpass the right-branching baseline, wherein the rightmost word functions as the head of the immediately adjacent left word, for grammatical structure induction.

Contemporary methods involve the utilization of neural networks (He et al., 2018) and LLM (Shen et al., 2021). Nevertheless, these innovative models may exhibit limitations when applied to languages with limited resources, particularly indigenous languages, and notably in the context of dependency grammar.

A noteworthy approach is the application of the MI measure, which has been harnessed to induce constituent grammar (Solan et al., 2005), and dependency relations for languages like Japanese (de Paiva Alves, 1996) and Portuguese (da Silva and Pardo, 2023).

Several initiatives have advanced in the domain of grammatical induction for languages with limited linguistic resources. Dahl et al. (2023) introduced a method employing Womb Grammars, a technique designed for the translational mapping of languages, in which grammar has been described to languages with no grammar description, to facilitate the induction of the Ch'ol language[3].

In what follows we present the data and the methods that we explore in this paper.

3. Methodology

We use data from UD version 2.13[4]. This contains 245 treebanks (i.e., corpora with sentences and their corresponding syntactical dependency analyses) for 141 languages. Almost 50 languages are indigenous or ethnic representative. Of these, twelve are spoken in Brazil and nine in Russia. The twelve languages used in this work are: Akuntsu, Guajajara, Kaapor, Karo, Makurap, Munduruku, Tupinamba, Nheengatu, Apurina, Bororo, Xavante, and Madi.

All these languages include 36,322 tokens, 8,632 types, and 5,000 sentences. A detailed description of these languages is presented in Table 1. The first column describes the language used in the experiment, the second shows the linguistic family, and the third column describes the number of different Syntactic Relations (SR) used in the annotations. Subsequent columns detail the number of tokens, vocabulary size, and complexity (computed as the type-token ratio). Higher complexity indicates greater sparsity. The final three columns present the number of sentences, the average number of tokens per sentence, and the standard deviation for token counting.

Nheengatu may stand out as the most extensively documented Brazilian indigenous language,

[2] https://github.com/diegodpgs/PROPORInd

[3] Ch'ol is an indigenous language of Mexico that lacks a formally documented grammar. However, it is noteworthy that the grammatical induction methodology articulated in this study relies on the use of syntactic relations, by definition, using supervised training.

[4] http://hdl.handle.net/11234/1-5287

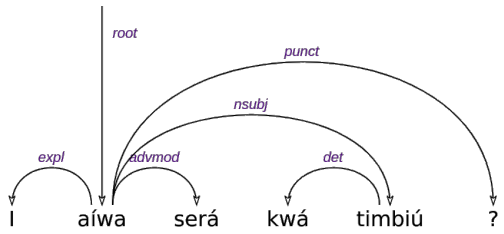


Figure 1: An example sentence for Nheengatu Language (Avila, 2021)

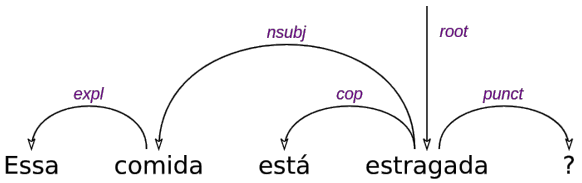


Figure 2: The translation to Portuguese of the sentence presented in Figure 1 (Avila, 2021)

dating back to its description in the first Brazilian indigenous language dictionary in 1756 (Avila, 2021). Moreover, numerous texts in Nheengatu were authored during the eighteenth century, further contributing to its rich documentation. The Nheengatu treebank is the largest one: 12,743 tokens (35% of all treebanks) and 1,913 types (22.1% of all treebanks). About 99.8% of all sentences have a length of up to 40 tokens (including punctuation), which is compared to almost all European languages available in UD initiative. For instance, in German, Czech and Russian, which are the biggest treebanks in UD, about 93% of sentences have a length of fewer than 40 tokens.

Since the UD repository only provides test sets, we perform cross-validation such that the test set is split into five folds: one for test and four for training. Three different grammar induction methods are used: MI, DMV, and LLM. In the present study, it is pertinent to emphasize that our approach is entirely unsupervised. Therefore, our training data solely comprises raw text, with the exception of the DMV method which incorporates gold Part of Speech (POS) tags.

The first works on grammar induction applied a dynamic programming algorithm on $O(n^3)$ for constituency grammar (Sankaran, 2010; Cohen et al., 2008), which is computationally expensive for longer sentences. For this reason, most works on grammar induction were trained on sentences up to 40 tokens (Kim et al., 2019) . In this paper, we tested the models on sentences of lengths up to 10 and up to 40 tokens, to evaluate the impact of different sentence size. The tree models used in this work are described in subsections 3.1, 3.2, and 3.3. These models are unsupervised, except for the LLM that, besides the zero-shot approach, we also used one and two-shot learning.

3.1. DMV Model

The DMV stands as a prevalent model for grammar induction, serving as a baseline in several works on unsupervised grammar induction (Shen et al., 2021; Yang et al., 2020). This model operates by generating syntactic trees in a top-down fashion using generative unsupervised training. The idea behind the DMV model is to estimate the syntactic tree by using the Expectation-Maximization (EM) algorithm. For each branch to be generated, it uses probability distributions to make decisions on when and which branch to generate.

We experimented with DMV using the same setting provided by He et al. (2018). It is pertinent to note that this model exhibits limitations in training with longer sentences, attributed to the $O(n^3)$ time complexity of the EM algorithm (Cohen et al., 2008; Spitkovsky et al., 2010). However, given the relatively small treebanks employed in this investigation, the DMV is executed with 10 epochs on each fold using cross-validation assessments.

3.2. MI-based Model

Generally defining, the MI measure indicates the dependency among elements of interest. In our case, it is used to determine words that are more probable to be syntactically related. Equation (1) shows how it is computed for head (h) words and their dependents (d).

$$MI(D,H) = \sum_{d \in D} \sum_{h \in H} P(d,h) log_2 \frac{P(d,h)}{P(d)P(h)} \quad (1)$$

To compute it, we performed word pair permutations within each sentence, considering every possible configuration. The total number of permuted pairs is described by $\sum_{d=1}^{DW} n - d$, where *n* is the number of tokens in the sentence, including punctuation, and *DW* is the distance between the words in the sentence. For instance, for the sentence *"I love the sun"*, the word pairs for DW=1 is <I, love>,<love, the>,<the sun>. Using DW=n, the number of pairs is described by binomial coefficients $\binom{n}{k}$, with *k* representing two (tokens per pair). This setting produces the pairs <I,love>,<I,the>,<I,sun>,<love,the>,<love,sun> and <the,sun>. We train all models using different DW values and choose DW=2 as the best performance.

That permutation process resulted in the creation of the final set of Sentence Permutations (SP), comprising pairs of tokens where the first token precedes the second in the sentence sequence. Following this, MI was computed for each word pair within the SP. Finally, we take the *n* pairs with the

Table 1: Indigenous languages in Brazil used in this study

Language	*Family*	*SR*	*Tokens*	*Types*	*Complexity*	*Sentences*	μ	σ
Xavante	Macro-Je	22	1,597	385	0.241	148	10.791	6.423
Tupinambá	Tupian	26	4,508	1,970	0.437	581	7.759	5.946
Nheengatu	Tupian	32	12,743	1,913	0.150	1,239	10.285	6.736
Munduruku	Tupian	26	1,022	399	0.390	158	6.468	5.977
Makurap	Tupian	15	178	95	0.533	37	4.811	1.998
Madi	Arawan	17	115	68	0.591	20	5.750	3.048
Karo	Tupian	25	2,319	773	0.333	674	3.441	1.523
Kaapor	Tupian	22	366	221	0.603	83	4.410	2.024
Guajajara	Tupian	27	9,160	1,515	0.165	1,182	7.750	4.041
Bororo	Bororoan	29	1,905	762	0.400	371	5.135	5.512
Apurina	Arawakan	26	941	373	0.396	152	6.191	3.258
Akuntsu	Tupian	21	1,468	506	0.344	343	4.280	2.556
All	-	35	36,322	8,632	4.208	5,000	7.264	5.450

```
1 Na sentença "Aiwana, paá, aintá uyaxiú",  as relações de dependência sintática
  são mostradas abaixo no formato (token dependente -> token cabeça)
2 (Aiwana -> uyaxiú)
3 (, -> paá)
4 (, -> paá)
5 (aintá -> uyaxiú)
6 (. ->  -> uyaxiú)
7 Liste as relações de dependência sintática na sentença "Yané tuixawa umanú ana
  mira amusuaxarawara usikié tenhẽ waá.", usando o formato (token dependente ->
  token cabeça).
```

Figure 3: An example of prompt for the Nheengatu language in one shot learning

highest MI and compare them to manually annotated sentences.

Since corpora used in this work are very small, we perform an edit distance smoothing. For each token in the test that was not in the training set, we searched for the most similar morphological token in the training set using edit distance. For instance, if the token *"uyapí"* does not appear in the training set, the edit distance is applied to find the most lexically related word in the training set, such as *"uyari"*. Then the frequency of the token *"uyari"* is assigned to the token *"uyapí"*. Since there will always be a lexically related token, all tokens in the test set will have a frequency. For bigrams found in the test and not in the training set, we apply a derived simple Laplace smoothing by attributing frequency equal to 1/size of the vocabulary.

3.3. Large Language Model

LLM are models that are trained with a massive amount of data and require a huge computational structure. They can be used in a wide number of tasks such as information extraction, summarization, and question answering, to name a few (Wei et al., 2022). We did not build the LLM using native languages, instead, since we do not have enough data, we used LLM trained in Portuguese. Since the native languages used in this work are spoken in Brazil, and their vocabularies eventually incorporate some Portuguese words, we believe that is possible to find some syntactic relations using LLM even if that language has never been used for training.

We aim to demonstrate the limits and potentialities of LLMs to learn syntactic information in languages with lower resources. We use the chatGPT 3.5 API provided by OpenAI. Differently from the experiments on MI and DMV, we select only three languages to conduct experiments with the LLM. As we wanted to analyze the influence of a larger treebank, we tested with Nheengatu. Average sentence length can also play a role in dependency grammar induction and, therefore, we chose the Karo language, whose sentences are shorter. Finally, we wanted to study the influence of the language family, and language Bororo was chosen for having the largest treebank among those languages not belonging to the Tupian family.

We performed zero, one shot, and two shots learning. In +1 shot learning, we use two different prompts: using a fixed sentence and a random sentence for composing the prompt. For the fixed sentence, we chose a sentence of length seven, which is approximately the average of all languages used in this study. The chosen sentence is the one with the most frequent tokens in the treebank. For the

prompt that applies a random sentence, we have random sentences with lengths up to 40 tokens in the training set to be included in the prompt. Since the answers provided by the model are not always the same, we tested the prompts on 30 sentences for each of the five folds of cross-validation. This experiment resulted in 2,250 requests to OpenAI API. We also tested different prompts in Portuguese language and chose the best one. An example of a prompt for one shot is shown in Figure 3.

4. Results

In this study, we adopt the 37 syntactic relations of the UD initiative[5], yet not all languages that we examined utilize all of these relations. As demonstrated in Table 1, Makurap employs only 15 syntactic relations, while Nheengatu utilizes 32. It is noteworthy that Guajajara does not include any occurrence of the subject relation. This study concentrates exclusively on syntactic relations that constitute a minimum of 10% of the respective treebank annotations. Due to limited data, we did not consider the subtypes of some syntactic relations.

We present results for the standard evaluation metrics: Undirected Dependency Accuracy (UDA) and Directed Dependency Accuracy (DDA). Comparing with the reference annotations, these metrics compute how many relations (for word pairs) were correctly predicted, considering or not the relation direction, respectively.

Overall, it is interesting that, despite the limited size of the treebanks, the induction methods for these languages achieved good results, even better than some reported results for non-indigenous languages, such as German, English, and Chinese, using DMV (Klein and Manning., 2004).

In general, Akuntsu and Karo emerged as languages exhibiting the best outcomes, whereas Guajajara and Xavante posed notable challenges. These results are not related to the family origin or annotation. Akuntsu, Karo, and Guajajara were annotated using the same annotation protocol within the same project (Gerardi et al., 2021). However, Akuntsu and Karo are two languages spoken in the state of Rondônia, but Guajajara and Kaapor, which are also spoken in the same state (Maranhão) and come from the same family, Tupian, present different outcomes.

No discernible correlation is observed between vocabulary size and treebank size; however, a subtle correlation is discerned between sentence length and associated scores. Across all settings, the "object" dependency relation was the most correctly detected one, yet substantial variation exists among languages.

MI presented the best results on UDA; on the other hand, DMV was better on DDA. As may be expected, LLM presented the worst results.

The syntactic relations that were more correctly induced (with the highest scores) with DMV are *punct* (punctuation) with 20.8%, *obj* (object) with 18.7%, and *nsubj* (subject) with 16.7%. However, MI presents the highest incidence of *obj* with 26% and *nsubj* with 18%, followed by *advmod* (adverbial modifier) with 8%. The selection of these syntactic relations is based on their prevalence within the treebank. Nonetheless, our code is accessible for retrieving data related to other syntactic relations as well.

The detailed results are presented in Subsections 4.1, 4.2, and 4.3. The summarized results are presented in Table 2. The last three lines present the most correctly induced syntactic relations (1 SR), the second most correctly induced syntactic relations (2 SR), and the third most correctly induced syntactic relations (3 SR), respectively. Due to space limitation, we presented only the results for DMV using the DDA metric[6].

Table 2: Summarized results

	DMV	MI	LLM
UDA 10	0.5135	**0.5692**	0.4165
UDA 40	0,.4654	**0.5089**	0.4212
DDA 10	**0.3201**	0.3122	0.2779
DDA 40	**0.2808**	0.1687	0.2720
1 SR	obj	obj	obj
2 SR	nsubj	nsubj	case
3 SR	punct	advmod	advmod

4.1. DMV

The results for DDA are presented in Table 3. DMV can induce correctly 89% of all object relations on Akuntsu, but only 11% on Kaapor. Despite presenting good results on small corpora such as those of Makurap and Madi, DMV struggles to induce some important syntactic relations. This pattern is similar when evaluated using UDA metrics.

4.2. MI

The use of edit distance yielded notable improvements, showcasing a 29.5% enhancement in MI for UDA and a 13.6% boost for DDA. While the results based on MI lag behind DMV in terms of DDA metrics, it is crucial to highlight the superiority of MI in UDA metrics. Moreover, it manifests superior outcomes in the context of induced object and subject relations. Notably, in the Makurap language, all object relations were accurately induced, and,

[5] https://universaldependencies.org/u/dep/index.html

[6] Detailed results may be found at https://github.com/diegodpgs/PROPORInd

Table 3: Results for DMV with DDA metric

Language		1 SR		2 SR		3 SR	
DDA for sentences $\leq$ 10 tokens							
Akuntsu	0.5661	0.8957	obj	0.5783	nsubj	0.5551	punct
Apurina	0.4248	0.7460	obj	0.7227	nsubj	0.2321	punct
Bororo	0.3832	0.8696	case	0.6992	obl	0.5489	nsubj
Guajajara	0.1669	0.4690	obl	0.2142	discourse	0.0730	punct
Kaapor	0.2500	0.7843	obj	0.4921	nsubj	0.1765	advmod
Karo	0.3803	0.5882	nsubj	0.4595	advmod		
Madi	0.4186	0.4545	punct	0.2500	obj		
Makurap	0.4696	0.6667	advmod	0.3750	discourse		
Munduruku	0.4074	0.8077	case	0.6846	obl	0.5000	punct
Nheengatu	0.3671	0.5756	advmod	0.5579	nsubj	0.2271	punct
Tupinamba	0.3138	0.5111	punct	0.4100	obl		
Xavante	0.3264	0.7500	dep	0.3099	punct	0.1176	nsubj
μ	**0.3729**	**0.6765**		**0.4795**		**0.3038**	
μ **weighted**	**0.3201**	**0.5907**		**0.4492**		**0.2193**	
DDA for sentences $\leq$ 40 tokens							
Akuntsu	0.5641	0.8800	obj	0.6077	nsubj	0.5879	punct
Apurina	0.3907	0.8488	obj	0.7211	nsubj	0.2153	punct
Bororo	0.3579	0.6647	punct	0.6497	obl	0.5020	nsubj
Guajajara	0.1704	0.4223	obl	0.2135	discourse	0.0900	punct
Kaapor	0.2287	0.8302	obj	0.4242	nsubj	0.2432	advmod
Karo	0.3301	0.5882	nsubj	0.4757	advmod		
Madi	0.3585	0.4167	punct				
Makurap	0.4348	0.6250	advmod	0.4375	discourse		
Munduruku	0.3784	0.9029	case	0.6506	nsubj	0.5909	obl
Nheengatu	0.2943	0.5376	nsubj	0.4918	advmod	0.1613	punct
Tupinamba	0.2572	0.4835	punct	0.3921	obl		
Xavante	0.3110	0.6348	dep	0.2800	punct		
μ	**0.3397**	**0.6529**		**0.4858**		**0.3415**	
μ **weighted**	**0.2808**	**0.5512**		**0.4198**		**0.1916**	

in the Madi language, every subject was correctly induced.

4.3. LLM

Differently from experiments with DMV and MI, we did not use weighted average for LLM because the Nheengatu language presents 75% of the available corpora. The results presented in Table 2 refer to the average of all settings. As we expected, the zero-shot for all languages and all settings yielded the least favorable results on average, with 0.290 for UDA and 0.142 for DDA; transitioning to one-shot learning, UDA improved to 0.413, and DDA to 0.264; in two-shot learning, the model achieved 0.427 for UDA and 0.285 for DDA. When sentences were not fixed, the model exhibited competence with scores of 0.431 for UDA and 0.286 for DDA. However, when fixed sentences were employed in the prompt for one and two-shot learning, the overall performance deteriorated, resulting in an average of 0.406 for UDA and 0.263 for DDA. This result may be due to the distribution of the sentences, since that, with no fixed sentence, almost 150 different sentences were tested in the prompt. However, to induce object relations, using a fixed sentence in the prompt presented better results.

Different from MI and DMV, LLM may be influenced by the size of the treebank. When using one and two-shot learning, Nheengatu presents 0.440 DDA, against 0.406 in Karo and 0,410 in Bororo. This result is different from the DMV and MI approaches, in which Nheengatu presents the poorest scores. Nonetheless, the induction of particular dependency relations may not necessarily exhibit a correlation with treebank size. In the cases of Karo and Bororo languages, accurate induction of object relations is achieved with notable proficiency. In contrast, the Nheengatu language demonstrates a lower level of accuracy in this regard. These outcomes align with the findings obtained through both DMV and MI approaches.

5. Discussion

Despite the effectiveness of modern approaches such as neural networks and LLM, simple methods such as MI can perform better when applied to low language resources. For some sentences, we

identified that the LLM likely employed the straightforward right-branching algorithm. It is necessary to note that an explicit evaluation of the comparative efficacy of these methodologies against the right-branching baseline, established at 0.38 for the English language (Klein and Manning., 2004), was not conducted and remains for future work.

The MI models present good results, but the induced syntactic tree could have missing elements, as presented in Appendix A. It can be solved by optimization, which could also be a matter of future work.

It is essential to highlight that the indigenous languages utilized in this study exhibit distinct syntactic characteristics, including the absence of certain crucial syntactic relations (such as nsubj in Guajajara, for example), as well as unique sentence structures. These nuances may influence the obtained outcomes. In-depth linguistic inquiries or even anthropological investigations may be necessary to elucidate the variations in results across different languages.

6. Final Remarks

We presented a study on grammar induction in different Brazilian indigenous languages. We demonstrate the efficacy of inducing syntactically related words for low-resource languages using simple approaches, mainly in inducing specific relations, such as object and subject relations. Such methods may be very useful to uncover syntactic structures for languages for which the grammar was not yet described or to refine NLP parsing methods. For future work, we aim to investigate other languages spoken in other countries.

Acknowledgments

This work was carried out at the Center for Artificial Intelligence of the University of São Paulo (C4AI - http://c4ai.inova.usp.br/), with support by the São Paulo Research Foundation (FAPESP grant #2019/07665-4) and by the IBM Corporation. The project was also supported by the Ministry of Science, Technology and Innovation, with resources of Law N. 8.248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published as Residence in TIC 13, DOU 01245.010222/2022-44.

References

Marcel Twardowsky Avila. 2021. *Proposta de dicionário nheengatu-português*. Ph.D. thesis, Universidade de São Paulo.

Marta Maria Azevedo. 2016. Urbanização e migração na cidade de são gabriel da cachoeira, amazonas. *Anais do XV Encontro Nacional de Estudos Populacionais*, pages 1–14.

Shay B. Cohen, Kevin Gimpel, and Noah A. Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, pages 321–328.

Aline Da Cruz. 2011. Fonologia e gramática do nheengatú: A língua geral falada pelos povos baré, warekena e baniwa. *Ph.D. Thesis,Vrije Universiteit Amsterdam*.

Diego Pedro Gonçalves da Silva and Thiago Alexandre Salgueiro Pardo. 2023. Induçao gramatical para o português: a contribuiçao da informaçao mutua para descoberta de relaçoes de dependência. In *Proceedings of the 14th Brazilian Symposium on Information Technology and Human Language*, pages 298–307.

Veronica Dahl, Gemma Bel-Enguix, Velina Tirado, and Emilio Miralles. 2023. Grammar induction for under-resourced languages: the case of ch'ol. In *Analysis, Verification and Transformation for Declarative Programming and Intelligent Systems: Essays Dedicated to Manuel Hermenegildo on the Occasion of His 60th Birthday*, pages 113–132. Springer.

Tiago Barbosa de Lima, André CA Nascimento, Pericles Miranda, and Rafael Ferreira Mello. 2021. Analysis of a brazilian indigenous corpus using machine learning methods. In *Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional*, pages 118–129.

Eduardo de Paiva Alves. 1996. The selection of the most probable dependency structure in japanese using mutual information. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 372–374.

Fabrício Ferraz Gerardi, Tiago Tresoldi, Carolina Coelho Aragon, Stanislav Reichert, Jonas Gregorio de Souza, and Francisco Silva Noelli. 2023. Lexical phylogenetics of the tupí-guaraní family: Language, archaeology, and the problem of chronology. *Plos one*, pages 1–25.

Richard Futrell, Peng Qian, Edward Gibson, Evelina Fedorenko, and Idan Blank. 2019. Syntactic dependencies correspond to word pairs with high mutual information. In *Proceedings of the 5th international conference on dependency linguistics*, pages 3–13.

Fabrício Ferraz Gerardi, Stanislav Reichert, and Carolina Coelho Aragon. 2021. Tuled (tupían lexical database): introducing a database of a south american language family. *Language Resources and Evaluation*, 55(4):997–1015.

K. David Harrison. 2008. When languages die: The extinction of the world's languages and the erosion of human knowledge. *Oxford University Press*.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised learning of syntactic structure with invertible neural projections. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1292–1302. Association for Computational Linguistics.

Jacob Louis Hoover, Wenyu Du, Alessandro Sordoni, and Timothy J. O'Donnell. 2021. Linguistic dependencies and statistical dependence. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2941–2963.

Yoon Kim, Chris Dyer, and Alexander M. Rush. 2019. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2369–2385. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. *Proceedings of the 42nd annual meeting of the association for computational linguistics*, page 478–485.

András Kornai. 2013. Digital language death. *PloS one*, pages 1–11.

Boda Lin, Zijun Yao, Jiaxin Shi, Shulin Cao, Binghao Tang, Si Li, Yong Luo, Juanzi Li, and Lei Hou. 2022. Dependency parsing via sequence generation. In *Findings of the Association for Computational Linguistics*, pages 7339–7353.

Sidney Facundes Moore, Denny and Nádia Pires. 1994. Nheengatu (língua geral amazônica), its history, and the effects of language contact. *Meeting of SSILA and the Hokan-Penutian Workshop*, pages 93–118.

Rosângela Morello. 2016. Censos nacionais e perspectivas políticas para as línguas brasileiras. *Revista Brasileira de Estudos de População*, pages 431–439.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Jan Hajič, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4033.

Claudio S Pinhanez, Paulo Cavalin, Marisa Vasconcelos, and Julio Nogima. 2023. Balancing social impact, opportunities, and ethical constraints of using ai in the documentation and vitalization of indigenous languages. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6174–6182.

Baskaran Sankaran. 2010. A survey of unsupervised grammar induction. *Manuscript, Simon Fraser University 47*, pages 1–63.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron C. Courville. 2021. Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 7196–7209.

Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. 2005. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, pages 11629–11634.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, pages 751–759.

Gerald Taylor. 1985. Apontamentos sobre o nheengatu falado no rio negro, brasil. *Amérindia: revue d'ethnolinguistique amérindienne*, pages 5–23.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Songlin Yang, Yong Jiang, Wenjuan Han, and Kewei Tu. 2020. Second-order unsupervised neural dependency parsing. In *Proceedings of the 28th International Conference on Computational Linguistic*, pages 3911–3924.

A. Illustration of grammar induction for Nheengatu

We present a sample of the induced relations for the sentence *Aikwé awá ururi indé u reyuri putari t ne rupí?*, which corresponds to *Was there anybody to bring you or did you yourself want to come?* in English, using DMV, MI, and LLM methods. The cited sentence represents a transcription of speech delivered by an indigenous Nheengatu speaker (Moore and Pires, 1994). It is important to note that the orthography utilized is not the original form, but has been adjusted to adhere to the UD framework.

In Figures 4, 5, and 6, the color orange means that the model correctly predicted the relation according the UDA measure (which does not evaluate the direction of the arrow), and green means that the model correctly predicted the direction too, as informed by the reference annotation (in Figure 7).

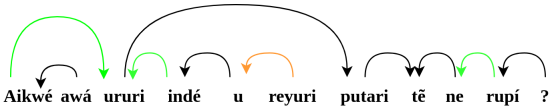


Figure 4: Induced relations using DMV

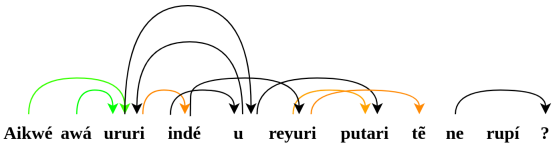


Figure 5: Induced relations using MI

Figure 6: Induced relations using LLM

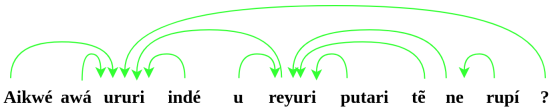


Figure 7: Reference annotation in the treebank

CAPÍTULO

7

AVALIAÇÃO EXPERIMENTAL

Os experimentos apresentados neste capítulo foram elaborados com o intuito de contribuir para as conclusões das hipóteses de pesquisa. Nos Capítulos 4, 5 e 6, são apresentados estudos que corroboram a utilidade da informação mútua na indução gramatical automática para o Português. Neste capítulo, uma análise mais detalhada é apresentada sobre a aplicação da informação mútua na língua portuguesa, em comparação com diferentes modelos.

Para responder às questões de pesquisa apresentadas no Capítulo 1.2, foi realizada uma análise de dados dos *treebanks* da língua portuguesa a fim de encontrar características da língua que poderiam ser úteis. Vários estudos que utilizam características peculiares de uma ou mais línguas já foram realizados. Spitkovsky, Alshawi e Jurafsky (2012) utilizaram a capitalização como marcador de relações sintáticas. A distribuição de classes de palavras foi implementada por Santamaria e Araujo (2010). A pontuação é outro marcador que pode ser utilizado para identificar relações sintáticas (SPITKOVSKY; ALSHAWI; JURAFSKY, 2011b).

Diferentemente dessas abordagens, foram utilizados dados sem nenhum tipo de anotação, nem mesmo anotação com as categorias morfossintáticas. Tamanho da palavra é um proxy para classes morfossintáticas abertas (substantivos, verbos, adjetivos e advérbios) e classes morfossintáticas fechadas (artigo, preposição, conjunção). Essa característica foi analisada em diferentes línguas e constatamos que para a língua portuguesa essa característica é mais relevante. Diante dessa descoberta, foi proposta a utilização de uma heurística simples, discutida na Seção 7.1. Isso resultou em uma melhoria considerável nos resultados da indução gramatical para o português ao utilizar a métrica DDA. Além disso, neste estudo, não foram encontrados marcadores sintáticos adicionais além dos apresentados em trabalhos anteriores que pudessem aprimorar o desempenho na tarefa de indução gramatical.

Na Seção 7.1, é apresentada a heurística utilizada neste estudo. Na Seção 7.2, a metodologia utilizada na experimentação dos diferentes modelos é apresentada. Finalmente, os resultados são apresentados na Seção 7.3.

7.1 Diferença de tamanho entre as palavras da relação

Um trabalho minucioso de análise dos *treebanks* foi realizado a fim de encontrar características que poderiam ser úteis para a tarefa de indução gramatical. Foram analisados a distribuição das classes gramaticais, a pluralização de palavras, a distribuição de sílabas e uso de *embeddings* como Word2vec (MIKOLOV *et al.*, 2013) e Glove (PENNINGTON; SOCHER; MANNING, 2014). No entanto, nenhum método se tornou tão eficaz quanto o uso da heurística de tamanho das palavras das relações.

Considerando a hipótese apresentada por Zipf (1949), categorias morfossintáticas fechadas, como determinantes, preposições e conjunções, tendem a ter um tamanho menor em comparação com classes morfossintáticas abertas, pois apresentam menor esforço.. Para investigar esses padrões, analisamos as 70 línguas com o maior número de *tokens* disponíveis no repositório da UD na versão 2.13 [1].

Nossa análise revelou que, em média, os determinantes na língua portuguesa possuem 1,692 letras por *token*, principalmente influenciada pelos artigos *o* e *a*. Considerando todas as línguas disponíveis no repositório UD, a língua portuguesa é classificada como a oitava língua com a menor média de comprimento na classe determinante. Além disso, entre as línguas que empregam o sistema de escrita alfabético ou cirílico, o português ocupa a terceira posição em termos de menor média, ficando atrás apenas do Húngaro, com 1,533 letras, e do Galês, com 1,560 letras.Em contraste, a média de comprimento dos determinantes em inglês é de 2,633 letras.

Tradicionalmente, determinantes, preposições e conjunções desempenham uma função de dependência na maioria das relações sintáticas em várias línguas. Isso é ilustrado na Tabela 5, que mostra a frequência de todas as categorias morfossintáticas em línguas que utilizam a UD e têm mais de 10 mil *tokens*. A primeira coluna apresenta as categorias morfossintáticas conforme definido pela UD. Na segunda e terceira colunas, são fornecidos o número total de palavras anotadas com cada categoria e a porcentagem dessas anotações como termo dependente, respectivamente. Na quarta e quinta colunas, são apresentados o total de termos e a porcentagem de palavras anotadas como termo cabeça, respectivamente.

Uma forma de exemplificar os resultados da Tabela 5 é por meio da análise da seguinte frase: “Esse número vem aumentando, uma vez que centenas de pessoas têm cruzado a fronteira todos os meses.”[2]. O artigo “uma” é termo cabeça na relação <uma, vez> e termo dependente na relação <cruzado, uma>. O verbo “cruzado”, além de ser termo cabeça na relação citada anteriormente, é também termo dependente na mesma frase na relação <aumentando, cruzado>.

Foi conduzida uma análise comparativa do tamanho dos termos das relações em várias

[1] <http://hdl.handle.net/11234/1-5287>

[2] Extraído do córpus Porttinari (PARDO *et al.*, 2021) disponível na versão 2.13 da UD com registro de sentença *FOLHA_DOC003143_SENT005*

Tabela 5 – Frequência de categorias morfossintáticas por relação de dependência

	Dependente		Cabeça	
UPOS	Total	Taxa	Total	Taxa
punct	3.824.200	99,70%	11.589	0,30%
adp	3.582.615	96,72%	121.640	3,28%
cconj	1.058.165	94,23%	64.839	5,77%
det	2.153.700	93,07%	160.242	6,93%
aux	1.253.641	90,66%	129.229	9,34%
sconj	628.986	90,12%	68.943	9,88%
part	408.527	87,14%	60.266	12,86%
adv	1.488.481	76,19%	465.171	23,81%
pron	1.473.925	73,46%	532.448	26,54%
sym	75.355	62,99%	44.281	37,01%
num	550.713	61,04%	351.569	38,96%
adj	2.002.889	56,89%	1.517.864	43,11%
x	172.511	53,58%	149.465	46,42%
intj	26.111	52,99%	23.161	47,01%
propn	1.594.748	48,47%	1.695.110	51,53%
noun	6.738.205	35,92%	12.018.508	64,08%
verb	2.039.483	14,89%	11.657.930	85,11%

línguas. Os resultados são apresentados na Tabela 6. A tabela está dividida em duas partes. À esquerda, estão listadas as 20 línguas em que a razão de tamanho entre os termos cabeça (C) e dependente (D) é maior, ordenadas de forma crescente. À direita, estão listadas as 20 línguas em que essa razão é menor, também ordenadas de forma crescente.

Na primeira coluna de cada segmento é apresentado o nome da língua. Em seguida, na segunda e terceira colunas são apresentadas as médias de tamanho do termo cabeça e do termo dependente da relação, respectivamente. Na última coluna do segmento é apresentada a proporção entre o tamanho do termo cabeça e o tamanho do termo dependente.

Os resultados indicam que, em uma amostra de mais de 70 idiomas, o Português está classificado em segundo lugar quando se considera a proporção entre o tamanho do termo cabeça e o tamanho do termo dependente na relação. É interessante observar que somente o Chinês clássico possui um termo dependente com uma média de tamanho maior que o termo cabeça. Idiomas que não adotam o sistema alfabético ou cirílico, como os principais idiomas da Ásia (chinês, japonês, árabe, urdu, hindi, entre outros), exibem uma proporção muito baixa, provavelmente devido aos seus sistemas de escrita, nos quais símbolos isolados podem apresentar semântica completamente diferente quando usados de forma conjunta com outros símbolos.

A descoberta mencionada pode ser compreendida através da lei de Zipf (ZIPF, 1949). Essa lei descreve a relação entre a frequência de uma palavra e sua posição na lista de palavras mais frequentes. Como ilustrado na Tabela 5, as categorias mais comuns incluem pontuação, preposição, conjunção e determinantes. No contexto deste estudo, as palavras mais frequentes são detalhadas na Tabela 7.

Tabela 6 – Tamanho médio dos *token* de cada termo da relação

Maior razão				Menor Razão			
Língua	C	D	razão	Língua	C	D	razão
Pomak	6.03	3.58	0.686	Islandês	5.51	4.06	0.356
Portugues	6.85	4.09	**0.674**	Sami nórdico	7.22	5.35	0.349
Francês antigo	5.55	3.34	0.664	Estoniano	7.06	5.31	0.330
Grego	7.55	4.56	0.655	Persa	4.90	3.69	0.328
Galego	7.22	4.36	0.654	Tâmil	8.41	6.38	0.319
Italiano	7.00	4.24	0.651	Manx	4.44	3.37	0.317
Catalão	6.76	4.11	0.646	Gótico	6.62	5.08	0.303
Espanhol	6.93	4.25	0.628	Filandês	7.92	6.13	0.291
Francês	6.63	4.12	0.611	Magrebino Francês Arábico	5.69	4.42	0.287
Hebreu antigo	7.07	4.43	0.597	Sâncristo	6.79	5.30	0.281
Eslovaco	6.85	4.32	0.587	Basco	6.83	5.56	0.227
Cóptico	3.98	2.52	0.579	Vietinamita	4.80	3.92	0.226
Alemão	8.56	5.47	0.566	Hindu	4.66	3.84	0.214
Holandês	7.12	4.62	0.542	Coreano	3.29	2.79	0.177
Búlgaro	6.88	4.47	0.538	Urdu	4.17	3.59	0.162
Africanês	7.53	4.90	0.538	Árabe	2.00	1.81	0.109
Ucraniano	7.03	4.57	0.538	Chinês	1.74	1.58	0.103
Esloveno	6.76	4.44	0.522	Japonês	1.20	1.11	0.080
Romeno	6.28	4.13	0.521	Chinês Clássico	1.02	1.05	-0.028

Tabela 7 – *Tokens* mais frequentes da língua portuguesa

token	Frequência
de	48,176
a	36,791
,	32,704
o	29,478
.	22,369
em	17,649
que	11,626
os	10,137
as	8,794

Conforme observado por Zipf (1949), é inerente à condição humana a busca pelo caminho de menor esforço. No âmbito da comunicação, palavras mais curtas demandam menos esforço em sua utilização, tanto em termos de tempo de pronúncia quanto de compreensão, embora o processo de aquisição da linguagem pelas crianças siga uma trajetória inversa, como apontado por Goodman, Dale e Li (2008). As palavras mais frequentes, como listadas na Tabela 7, muitas vezes carecem de significado por si só, dependendo de outros termos para transmitir sentido na comunicação. Por outro lado, palavras mais longas, por geralmente serem de classes abertas e, portanto, apresentarem uma semântica mais difusa, tendem a assumir com maior frequência o papel de termo cabeça em uma relação, como pronomes próprios (PROPN) com 51%, substantivos(NOUN) com 64% e verbos(VERB) com 85% , como demonstrado na Tabela

5.

Com base nas descobertas, foi adotada a heurística apenas ao considerar a direção da dependência na relação DDA. Em todos os casos, sem distinção, a maior palavra foi considerada como o termo cabeça e a menor como o termo dependente nos experimentos com informação mútua. Essa heurística também foi incorporada ao modelo neural, resultando em melhorias de desempenho em ambos os casos.

7.2 Metodologia

Utilizamos os *treebanks* disponibilizados na UD na versão 2.13[3] em língua portuguesa. Optamos por selecionar *treebanks* que apresentam melhor uniformidade na anotação: Petrogold, Porttinari e Bosque. Todos os três *treebanks* foram combinados em um único conjunto de dados, uma vez que este estudo não se concentra na análise das peculiaridades de cada *treebank*. A Tabela 8 apresenta uma análise estatística dos dados. Na primeira linha, é indicado o número de sentenças por conjunto de dados (treino e teste). Em ambos os conjuntos há sentenças de todos os treebanks utilizados neste estudo. O conjunto de treino representa 84.8% de todas as sentenças, enquanto o conjunto de teste representa 15.2%. Na segunda linha, é apresentado o número de *tokens* em cada conjunto de treino e de teste. Nas terceira e quarta linhas, são fornecidas a média e o desvio padrão de *tokens* por sentença, respectivamente. A maior sentença é indicada na linha cinco, e os 5º e 95º percentis do número de *tokens* por sentença são apresentados nas linhas subsequentes. Observa-se que a distribuição de palavras entre os conjuntos de teste e treino é bastante semelhante.

Tabela 8 – Principais estatísticas

	Treino	Teste
Sentencas	20,083	3,888
Tokens	488,515	90,795
μ T/S	24.3248	23.3526
σ T/S	15.186	13.9861
Maior sentença	239	123
5 percentil	5	6
95 percentil	52	50

Na Tabela 9, são apresentadas as seis relações sintáticas mais comuns tanto para o conjunto de teste quanto para o de treino. Observa-se que as relações sintáticas CASE e DET são bastante frequentes. Na relação sintática DET, em um dos dois termos na relação, sempre haverá um determinante, contribuindo assim com a alta frequência. Já a relação sintática CASE, um dos dois termos sempre será uma preposição. Conforme apresentado na Tabela 5, preposições e determinantes estão entre as categorias morfossintáticas mais frequentes.

[3] <http://hdl.handle.net/11234/1-5287>

Tabela 9 – Relações sintáticas mais frequentes

	Treino		Teste	
1	CASE	0.1463	DET	0.1479
2	DET	0.1450	CASE	0.1461
3	PUNCT	0.1256	PUNCT	0.1256
4	NMOD	0.0900	NMOD	0.0889
5	OBL	0.0554	OBL	0.0543
6	NSUBJ	0.0508	NSUBJ	0.0533

No pré-processamento realizado neste estudo, nenhum *token* foi removido. Para diminuir a esparsidade dos dados, foram mantidas todas as letras em caixa baixa. As palavras contraídas, por exemplo, preposição + artigo (em + o = no), foram removidas do *treebanks* por representarem duplicidade com as palavras em separado sem a contração. Essas palavras estão identificadas com ID X-(X+1) nos *treebanks* utilizados, sendo X o número de identificação do *token* na sentença. Nos modelos DMV e de redes neurais, adotamos o mesmo processo de pré-processamento utilizado pelos autores desses estudos que estamos reproduzindo, o qual inclui a eliminação de pontuação.

Os primeiros trabalhos em indução gramatical utilizavam o algoritmo EM, a qual tem uma ordem de execução assintótica de $O(n^3)$ (SANKARAN, 2010; COHEN; GIMPEL; SMITH, 2008). O modelo DMV, assim como o modelo neural implementado neste estudo, utiliza este algoritmo. Isso corrobora para um tempo de execução e consumo de memória bastante expressivos. Por este motivo, os modelos DMV e Neural foram treinados utilizando sentenças de até 30 palavras. Para os demais modelos, IM e modelos de língua, foram realizados apenas um único experimento com a sentenças de até 100 palavras. Com base neste experimento, é viável distinguir os resultados para sentenças de qualquer extensão, desde que não ultrapassem 100 palavras. Nas subseções subsequentes, forneceremos uma descrição mais detalhada da metodologia para cada modelo.

7.2.1 IM

Foi implementada a mesma configuração utilizada no trabalho apresentado no Capítulo 4. Optamos por aplicar permutação de palavras com uma distância de até dois, e selecionamos os N pares com a maior informação mútua, onde N representa o número de *tokens* da frase. A métrica DDA foi calculada de duas maneiras distintas. Na primeira abordagem, considera-se a ordem de aparição dos termos na frase, onde o termo dependente é posterior ao termo cabeça. Na segunda abordagem, o critério de definição do termo cabeça e do termo dependente é baseado no tamanho das palavras, sendo o termo cabeça a palavra maior e o termo dependente a palavra de menor tamanho na relação.

Foram ainda implementados dois métodos de suavização: distância de edição e suavização de Laplace + 1. Nesse último método, durante o cálculo das probabilidades, adiciona-se

1 ao numerador e o tamanho do vocabulário ao denominador. Este procedimento impede que termos não observados durante o treinamento resultem em informação mútua igual a zero. A configuração desses dois métodos foram as mesmas utilizadas no trabalho apresentado no Capítulo 4.

7.2.2 DMV

Foi utilizada a mesma configuração utilizada no trabalho apresentado no Capítulo 4. O trabalho em que o treinamento do DMV foi reproduzido, faz uma conversão de gramática de constituintes para gramática de dependência (HE; NEUBIG; BERG-KIRKPATRICK, 2018). No presente estudo, a gramática de dependência é utilizada diretamente no modelo sem a necessidade de conversão. Por esse motivo, além da diferença de línguas, a comparação dos resultados deste trabalho com o trabalho original reproduzido não é direta.

7.2.3 Neural

Com base na revisão sistemática apresentada no Capítulo 3, foram encontrados apenas dois estudos que implementam redes neurais para gramática de dependência sem usar categorias morfossintáticas (HE; NEUBIG; BERG-KIRKPATRICK, 2018; YANG; ZHAO; TU, 2021). Foi selecionado o estudo (HE; NEUBIG; BERG-KIRKPATRICK, 2018) por apresentar melhor performance em gramática de dependência. Com base na revisão sistemática conduzida nesta dissertação, o presente estudo é o primeiro trabalho que aplica redes neurais em gramática de dependência diretamente em texto puro.

O modelo neural utilizado neste trabalho etiqueta as categorias morfossintáticas de forma não supervisionada usando HMM. Posteriormente, as categorias morfossintáticas são utilizadas para gerar a saída do modelo DMV que, finalmente, servem de entrada para o modelo Neural. O modelo utiliza *embeddings* no processo de treinamento. Para o treinamento, foram aplicados duas estratégias: o uso de *embeddings* pré-definidos e *embeddings* construídos apenas com os dados de treino e teste. Esta distinção foi necessária para averiguar se a forma que os *embeddings* são treinados influencia o resultado final do modelo. No total, foram realizados 12 experimentos para diferentes combinações de *embeddings* e uso da heurística descrita na Seção 7.1.

Foram utilizados o Word2Vec (MIKOLOV *et al.*, 2013) e Glove (PENNINGTON; SOCHER; MANNING, 2014) como entrada do modelo neural. Em ambos experimentos foram utilizados vetores de 100 dimensões. O estudo original também implementa Skip-Gram utilizando dimensão de tamanho 100 treinados em 1 bilhão de tokens. Dois diferentes modelos de *embeddings* foram utilizados neste trabalho com o objetivo de avaliar se os diferentes modelos construídos de diferentes formas poderiam influenciar no resultado final. Os *embeddings* já treinados utilizados neste estudo utilizam mais de 1 bilhão de *tokens* (HARTMANN *et al.*, 2017)

[4], e os modelos construídos com os *tokens* de treino e teste totalizam 579,310 *tokens*.

Adicionalmente à essa configuração, foram combinados os *embeddings* treinados com dados de treino e teste dos *treebanks* e os *embeddigns* pré-treinados. Esta combinação é feita inicialmente verificando se a palavra está contida nos *embeddings* construídos com os dados de treino e teste. Se a palavra não foi vista no treinamento, é realizada uma procura nos *embeddings* treinados com 1 bilhão de *tokens*. Caso a palavra não seja encontrada, um vetor de zeros é apresentado como entrada para a respectiva palavra. Estas combinações totalizaram 12 experimentos diferentes. Apenas a versão pré-treinada do modelo Glove foi utilizada.

7.2.4 Grandes modelos de língua

Foram utilizados dois diferentes modelos de língua, o GPT-3 (BROWN *et al.*, 2020) e o Sabiá (PIRES *et al.*, 2023) que é baseado no LLaMA (TOUVRON *et al.*, 2023). O *ChatGPT* ganhou bastante popularidade nos últimos anos devido aos seus resultados satisfatórios em várias tarefas de PLN (LASKAR *et al.*, 2023). Por essa razão, optamos por utilizar o GPT-3 nesse trabalho. A escolha do Sabiá deve-se ao fato de ser um modelo treinado exclusivamente em língua portuguesa e demonstrar um bom desempenho em várias tarefas, algumas das quais superam o GPT-3 (PIRES *et al.*, 2023).

Para o GPT-3 foi utilizada a API disponibilizada pela Open AI para o uso automatizado do chatGPT[5] (ABDULLAH; MADAIN; JARARWEH, 2022). Para o Sabía, foi utilizada a biblioteca MariTalk[6] disponibilizada pela Matiraca Ai. Para ambos os modelos de língua, foi utilizada a mesma configuração de treinamento, inclusive o mesmo *prompt*, aplicada no trabalho apresentado no Capítulo 4.

7.3 Resultados

Nesta seção são apresentados os resultados dos experimentos usando os diferentes modelos. Os resultados são apresentados separadamente por modelos e posteriormente a melhor configuração de cada modelo. Os resultados foram apresentados usando gráficos de linha para o desempenho acumulado por tamanho de sentença. Esta abordagem foi necessária para que seja possível compreender o desempenho do modelo com diferentes tamanhos de sentenças. A estabilidade em todos os gráficos é observada quando as sentenças ultrapassam os 25 *tokens*, devido à escassez de sentenças longas, o que resulta em um impacto menor no resultado acumulado à medida que o tamanho das sentenças aumenta. Para todos os modelos foi verificado o desempenho nas duas relações sintáticas mais importantes, nsubj e obj, e nas relações sintáticas mais frequentes nos *treebanks* utilizados, det e case.

[4] http://www.nilc.icmc.usp.br/embeddings
[5] https://chat.openai.com/
[6] https://chat.maritaca.ai/

Na Seção 7.3.1, são apresentados os resultados para os modelos de língua Sabiá e GPT3. Na Seção 7.3.2, são apresentados os resultados utilizando redes neurais, que implementa o modelo DMV como entrada do modelo neural. Na Seção 7.3.3, são apresentados os modelos de MI e DMV. Finalmente, na Seção 7.3.4, é apresentada a melhor configuração de cada modelo.

7.3.1 Grandes modelos de língua

Aproximadamente 99.5% das requisições feitas ao chatGPT foram atendidas conforme o padrão apresentado pelo *prompt*, enquanto o Sabiá obteve uma taxa de sucesso de 99.7% das requisições satisfeitas. A requisição não é considerada atendida quando não há resposta da API, seja por problemas na rede, limitações da própria API, ou quando a requisição não pode ser convertida para o padrão exigido, que é então descartada. Por exemplo, caso a API retorne relações em alguns dos seguintes formatos não presentes no prompt: *"termo1,termo2", "termo1 - termo2", "termo1 → termo2"* e *"termo1 - termo2"*, estes são convertidos para o formato padrão (termo1 → termo 2).

O Número de Relações Sintáticas Geradas (NRSG) a partir de sentença sem anotação usando modelos de língua apresenta um desvio padrão considerável em relação ao Número das Relações Sintáticas Anotadas (NRSA) por humanos. O NRSG a partir da sentença anotada são maiores que NRSA em 3% para *shot* zero, 6% para *shot* um e 10% para *shot* dois. Por exemplo, a frase "Ainda não sabemos a onde esse populismo nos levará. "[7] tem NRSA=10 enquanto o chatGPT tem NRSG=13: (as → sabemos), (ainda → sabemos), (não → sabemos), (sabemos → root), (sabemos → levará), (levará → sabemos), (a → onde), (onde → levará), (esse → populismo), (populismo → levará), (nos → levará), (levará → levará), (. → levará).

No modelo Sabiá, o *shot* zero apresentou uma taxa de 1.1% para requisições em que NRSG era maior que NRSA, 2.8% para *shot* um e 1.7% para *shot* dois. Ao considerarmos NRSG menor que NRSA, o chatGPT gerou 31.1% para *shot* zero, 21.2% para *shot* um e 16.1% para *shot* dois. Para o modelo Sabiá, a taxa foi significativamente superior ao chatGPT, com 51.4% das requisições obtendo NRSG < NRSA para *shot* zero, 25.7% para *shot* um e 24.9% para *shot* dois. Essas constatações evidenciam uma das limitações dos modelos de linguagem na identificação de relações sintáticas em uma sentença.

Apesar de o quantitativo de NRSG < NRSA ser maior que o quantitativo de NRSG > NRSA, os modelos ainda apresentaram um bom resultado. Esta diferença pode ser percebida se comparada entre os dois modelos: GPT e Sabiá nas Figuras 16 e 17. Uma vez que a acurácia é computada a partir da interseção entre o conjunto de relações sintáticas geradas e o conjunto de relações sintáticas anotadas, naturalmente que o Sabiá tende a apresentar menor resultado que GPT por ter menos relações geradas. Observa-se que em ambos os modelos, o desempenho do *shot* dois é maior que *shot* um, que, por sua vez, é maior que o *shot* zero.

[7] Extraído do córpus Porttinari disponível na versão 2.13 da UD com ID de sentença **FOLHA_-DOC000163_SENT020**

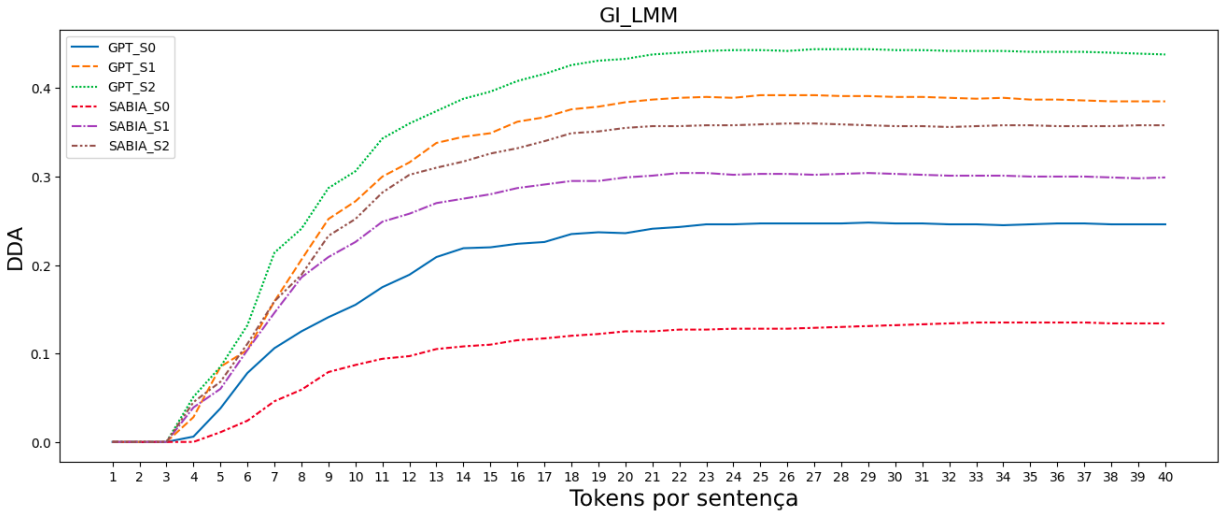


Figura 16 – Métrica DDA para os modelos de língua

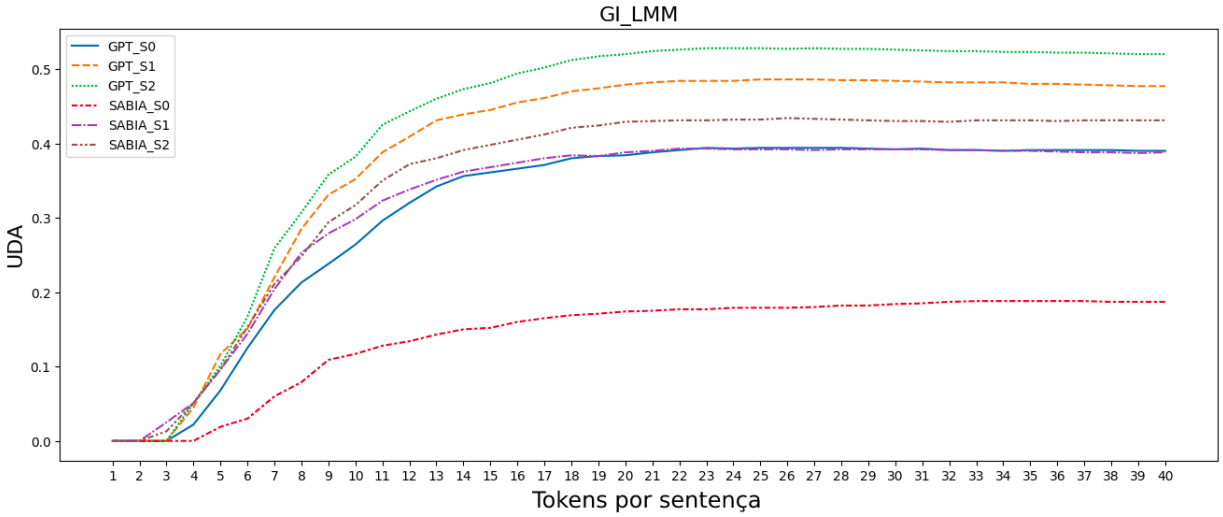


Figura 17 – Métrica UDA para os modelos de língua

Tabela 10 – UDA para diferentes relações sintáticas

	GPT S0		GPT S2		Sabiá S0		Sabiá S2	
	10	30	10	30	10	30	10	30
nsubj	0,369	0,441	0,494	0,560	0,144	0,196	0,431	0,443
obj	0,696	0,719	0,725	0,811	0,284	0,213	0,637	0,677
case	0,472	0,539	0,565	0,574	0,206	0,216	0,488	0,556
det	0,372	0,466	0,682	0,707	0,257	0,355	0,487	0,565

Na Tabela 10, são apresentados o desempenho para a métrica UDA usando os modelos GPT e Sabiá para zero (S0) e dois shot (S2) considerando relações sintáticas. Foram selecionadas as duas relações sintáticas mais importantes (nsubj e obj) e as duas relações sintáticas mais frequentes (case e det). Observa-se que, mesmo com zero shot, foi possível obter uma alta acurácia para a relação sintática obj, que na UD refere-se à relação de objeto, a partir do GPT. A

relação sintática nsubj apresenta o menor desempenho em todas as configurações. Esta diferença de desempenho para nsubj talvez esteja relacionada com a distância entre os termos da relação na sentença. Em aproximadamente 33% das relações nsubj, a distancia entre os termos da relação é maior que três na sentença. Por outro lado, obj, case e det apresentam respectivamente, 9%, 1.4% e 0.1% casos com distancia maior que 3 nas relações sintáticas em que esses ocorrem.

7.3.2 Redes Neurais

Os resultados demonstraram que nenhuma das 12 configurações de experimentos diferentes apresentaram diferença estatística significativa e, numericamente, a diferença foi muito baixa, apenas 0.005. Aparentemente, o montante de dados utilizado no treinamento, o tipo de modelo, e a combinação entre eles não apresentaram impacto no resultado. Isso sugere que provavelmente o modelo neural aplicado é fortemente dependente do modelo DMV.

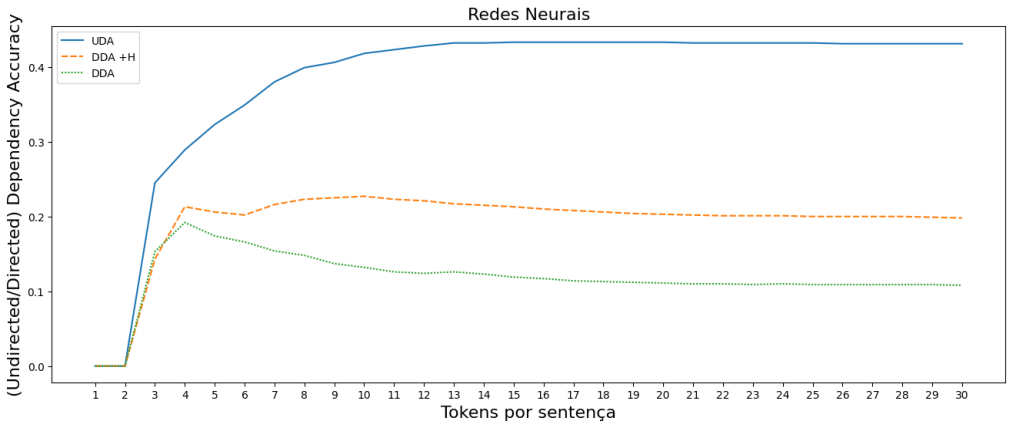


Figura 18 – Desempenho de IM em redes neurais

Na Figura 18, é apresentado o desempenho para as métricas UDA e DDA (com e sem heurística). Observa-se que o desempenho da métrica DDA para redes neurais é muito baixo. Este modelo aplicado para a língua inglesa apresenta resultados, no trabalho original, acima de 0.40 DDA em diferentes configurações. Isso pode ter ocorrido por diversos motivos que não foram verificados neste trabalho: conversão de uma gramática de constituinte para gramática de dependência, a não implementação do modelo usando a formalização da UD, se ocorre *overfitting* em relação aos dados da *WSJ* e, por fim, se o modelo funciona melhor para a língua inglesa em comparação com outras línguas. Essas hipóteses não foram testadas neste trabalho de mestrado, o que pode ser tema para trabalhos futuros.

A Tabela 11 exibe os resultados de diversas relações sintáticas para diferentes configurações de *embeddings*. O prefixo UD indica os *embeddings* treinados exclusivamente com dados do repositório UD, enquanto OD refere-se aos *embeddings* provenientes de outras fontes de dados. Na primeira coluna, são apresentados os resultados para os *embeddings* Skip treinados utilizando apenas os dados disponíveis da UD. Na segunda coluna, são apresentados os resultados dos

embeddings Skip-UD ∪ Skip-OF. Os dados apresentados nas colunas três e quatro correspondem, respectivamente, aos *embeddings* dos modelos Glove e Skip treinados com outras fontes de dados.

Observa-se que os resultados de uma mesma relação sintática são praticamente idênticos para diferentes configurações. A relação sintática det apresenta um desempenho muito baixo, provavelmente devido a sua proximidade na sentença com diferentes categorias morfossintáticas, uma vez que os *embeddings* são construídos a partir de um tamanho N de janela.

Tabela 11 – Resultados para algumas relações sintáticas com diferentes configurações de embeddings

	Skip-UD		Skip-UD∪ Skip-OF		Glove-OF		Skip-OF	
	10	30	10	30	10	30	10	30
nsubj	0.465	0.519	0.465	0.519	0.465	0.518	0.465	0.519
obj	0.299	0.336	0.299	0.336	0.299	0.336	0.299	0.335
case	0.606	0.601	0.606	0.600	0.606	0.601	0.606	0.601
det	0.135	0.126	0.135	0.126	0.135	0.125	0.135	0.125

7.3.3 IM vs DMV

O DMV é o modelo clássico de IG que utilizamos como *baseline*, portanto, nessa seção, apresentamos os resultados tanto do modelo DMV quanto do modelo proposto neste estudo, o IM.

Foi implementada a mesma configuração utilizada no trabalho apresentado no Capítulo 4. Apesar de o modelo ter sido implementado para línguas indígenas, a distribuição de frequência de distâncias é bem próxima. É importante informar que nos *treebanks* utilizados neste trabalho, 63.5% das relações apresentam distância de até duas palavras dentro da frase entre os termos da relação. Portanto, o teto desta configuração é de 0.635 DDA e 0.635 UDA. Foi realizado experimentos com distâncias de até 10 (93.4% de todas as relações). No entanto, uma vez que a esparsidade aumenta, a chance de o modelo encontrar uma relação sintática corretamente diminui drasticamente. Para calcular o resultado final deste modelo, foi utilizado o número total de relações, e não os 63.5% de relações possíveis.O modelo IM obteve seu melhor resultado com uma pontuação de 0.44 para UDA. Portanto, podemos concluir que, das relações possíveis, 69.2% foram identificadas.

A Figura 19 apresenta os resultados acumulados considerando o tamanho das sentenças. Os resultados demonstram que o modelo de IM e DMV têm alta correlação. Ainda observa-se que, estatisticamente, o desempenho é igual entre o modelo de IM que usa distância de edição com o modelo DMV. No entanto, numericamente, o DMV apresenta pior desempenho UDA para sentenças longas. Na Tabela 12, os números para sentenças de tamanhos até 10 e até 30 são apresentados. Na primeira coluna, apresenta-se o número máximo de *tokens* por sentença utilizado no treinamento. As colunas consequintes apresentam o desempenho UDA para IM usando distância de edição como suavizaçaõ, IM usando suavização de Laplace, IM

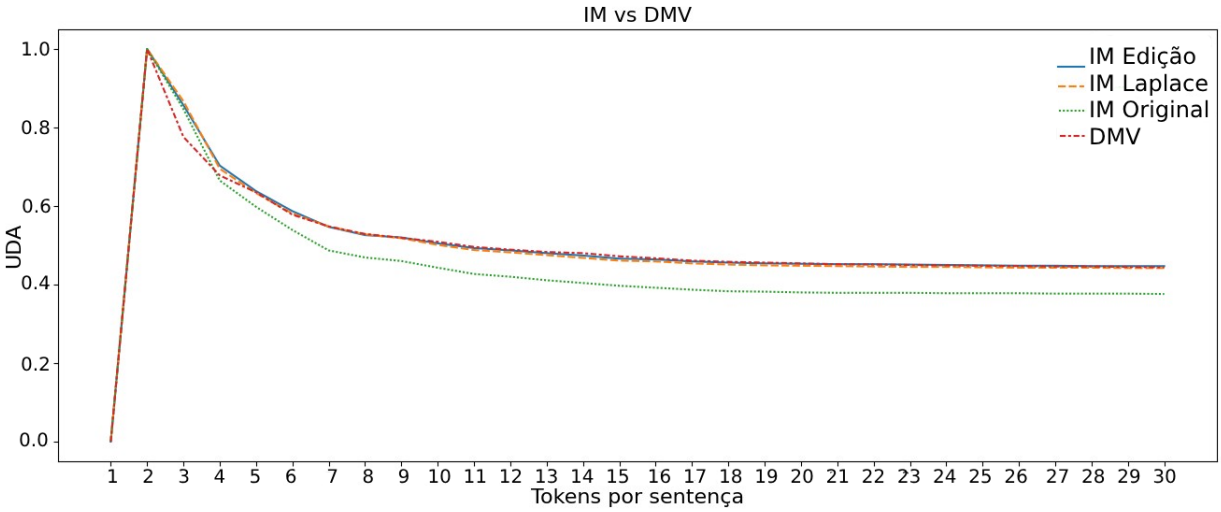


Figura 19 – Comparação de modelos IM e DMV usando a métrica UDA

sem suavização e o algoritmo DMV, respectivamente. Observa-se que a aplicação da suavização de distância de edição no modelo IM resultou em uma melhoria razoável de 5% no DDA em comparação com o modelo sem suavização. Por outro lado, em relação à suavização de Laplace, a melhoria foi de apenas 4%.

Tabela 12 – UDA para DMV e IM

tokens	IM Edit	IM Laplace	IM Original	DMV
10	0,493	0,488	0,469	**0,496**
30	**0,447**	0,442	0,376	0,444

Na Figura 19, é apresentado os resultados para a métrica UDA. Diferentemente da métrica DDA, o modelo IM apresenta resultado superior numericamente ao DMV quando utilizando o distância de edição e suavização. O DVM apresenta o pior resultado dentre os modelos apresentados.

Na Figura 20, são apresentados os resultados para a métrica DDA usando os métodos IM e DMV. Observa-se, que para sentenças curtas de tamanho até 6, o DMV e IM usando distância de edição suavizaçaõ apresentam resultados interessantes.

7.3.4 Discussão

A tarefa de induzir gramática a partir de texto não anotado de forma completamente supervisionada exige grande complexidade. Conforme apresentado no capítulo 3, há modelos que apresentam resultados superiores aos apresentados neste capítulo. No entanto, estes trabalham com anotação de classes morfossintáticas ou não são aplicados para gramática de dependência.

Os resultados que utilizaram grandes modelos de língua claramente apresentam vantagens em relação aos demais modelos. No entanto, estes resultados são positivos apenas para dois *shot*,

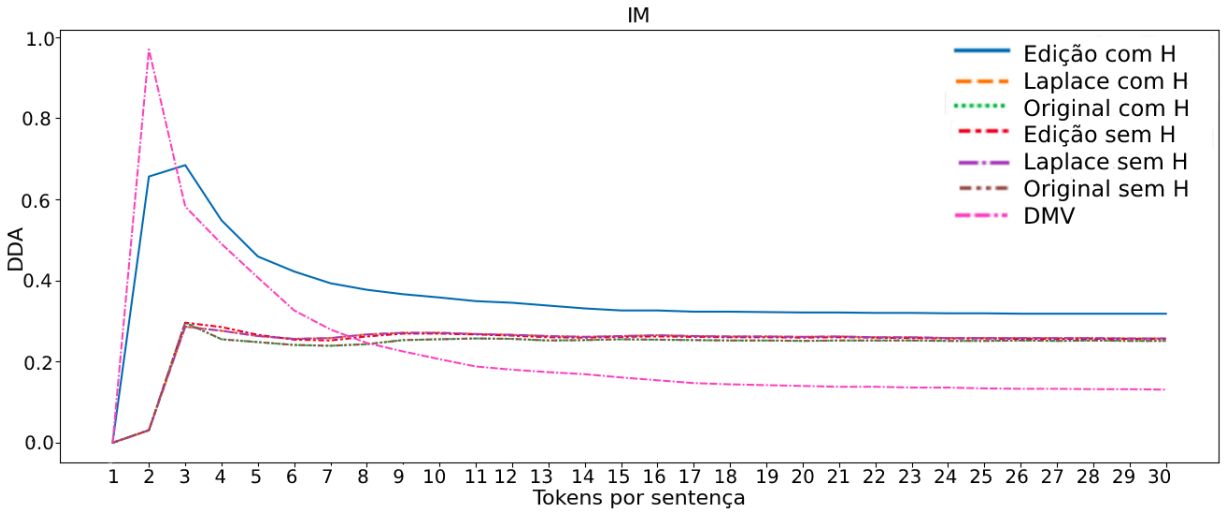


Figura 20 – Métrica DDA com IM e DMV

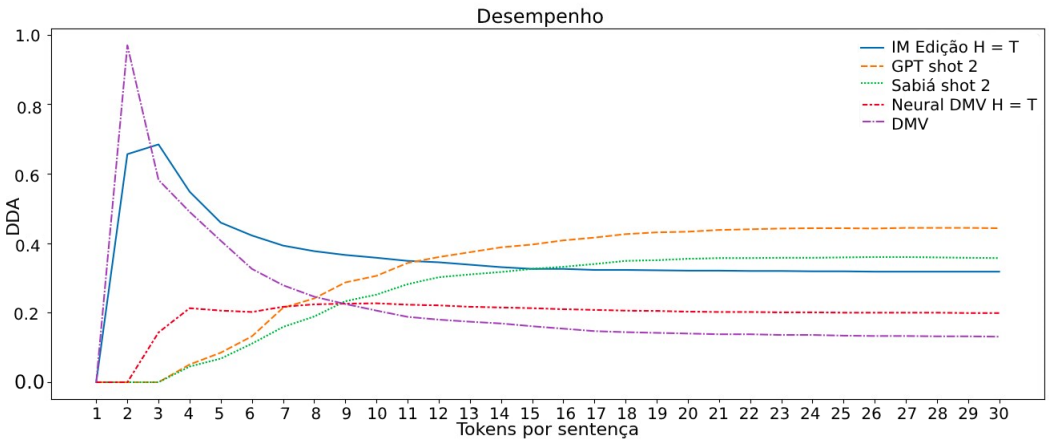


Figura 21 – Métrica DDA para todos

quando dois exemplos são apresentados, o que de certa forma, não se trata de um treinamento semi supervisionado. Já o modelo neural apresenta dificuldades por depender do modelo DMV. O modelo IM apresenta uma alta correlação com o modelo DMV. Isso sugere, que uma vez tanto o modelo IM quanto o modelo DMV são modelos probabilísticos, enfrentam problemas similares de esparsidade.

Nas Figuras 21 e 22, são apresentados os resultados compilados. Observa-se que os modelos DMV e IM apresentam melhores maior vantagem em relação à modelos de língua quando utilizado à métrica UDA. Os modelos DMV e IM são superiores que aos modelos de língua, inclusive com *shot* dois até sentenças de tamanho 14. Por outro lado, para a métrica DDA, os modelos de língua mostraram-se superiores com resultados melhores que o IM e DMV a partir de sentenças maiores que 11. Percebe-se ainda que o uso de edição de distância e heurística de tamanho dos termos da relação contribuem para um maior distanciamento de peformance entre IM e DMV, a medida que o número de sentenças aumentam. Numericamente, o modelo

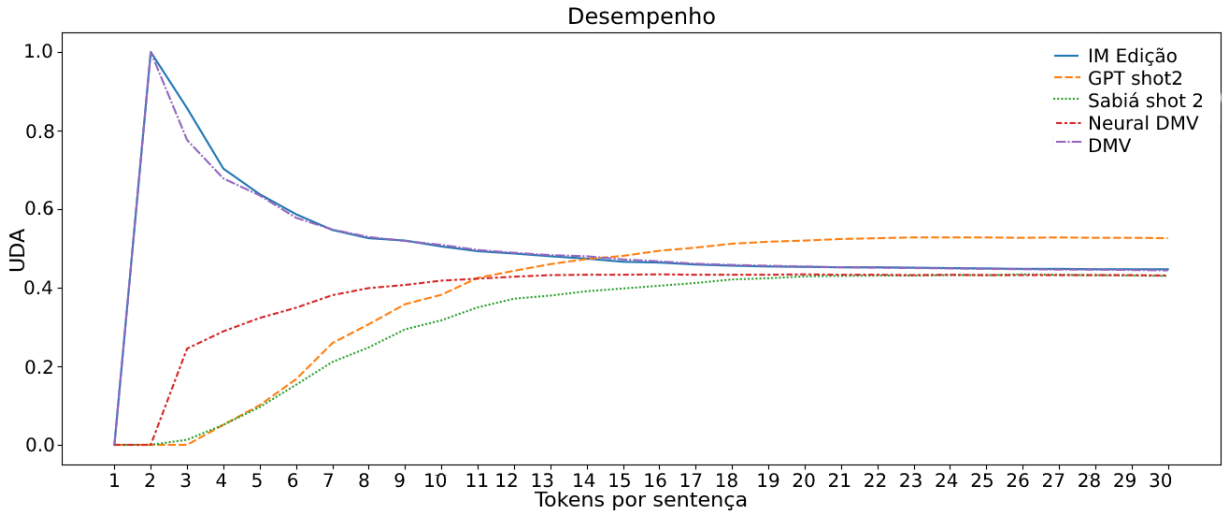


Figura 22 – Métrica UDA para para todos

IM apresentado neste trabalho tem desempenho numericamente inferior apenas ao GPT shot 2.

O estudo realizado neste capítulo demonstra que IM pode apresentar melhores resultados em relação a modelos muito mais complexos na tarefa de indução gramatical.

CAPÍTULO

8

CONCLUSÕES

O objetivo geral neste trabalho foi desenvolver, e avaliar métodos de indução gramatical especificamente para a língua Portuguesa com a finalidade de obter melhor desempenho que os modelos atuais. Esta tarefa requer alguns desafios importantes. O primeiro se refere ao extenso campo da indução gramatical que inclui não apenas a área de computação, mas também bioinformática, psicologia e linguística. O segundo problema é devido à pequena quantidade de sentenças anotadas com relações de dependência para a língua portuguesa. Por fim, a carência de trabalhos na área voltados para a língua portuguesa dificultaram os trabalhos de pesquisa.

Para resolver o primeiro problema, foi realizado um vasto mapeamento sistemático a fim de permitir estudar os principais modelos empregados na tarefa de indução gramatical, como também distinguir os modelos empregados para computação e os voltados para as demais áreas. Para enfrentar o segundo problema, foi proposta o uso de edição de distância como técnica de suavização para mitigar a carência de dados. Por fim, a carência de trabalhos de indução gramatical para a língua portuguesa foi mitigada a partir do estudo de métodos e modelos que empregaram a língua portuguesa. A partir destes estudos, foi possível compreender a efetividade de algumas técnicas empregadas na língua portuguesa.

Neste trabalho, buscou-se responder à duas questões de pesquisa: há características específicas da língua portuguesa que contribuem para um melhor desempenho em tarefas de IGNS *(i)*. A segunda, se é possível desenvolver técnicas de IGNS para a língua portuguesa melhores que os sistemas generalistas atuais *(ii)*. Para responder a primeira pergunta, foi considerado o tamanho dos termos da relação para distinguir o termo cabeça e o termo dependente. Essa heurística apresentou DDA de 0.316, para a melhor configuração, usando informação mútua. Este valor representa um acréscimo de 25% em relação ao mesmo método sem uso da heurística.

Segundo o mapeamento sistemático apresentado no Capítulo 3, foram encontrados apenas quatro trabalhos sobre indução gramatical em gramática de dependência que utilizam texto sem nenhum tipo de anotação. Estes estudos são descritos abaixo;

- Unsupervised Learning of Syntactic Structure with Invertible Neural Projections (HE; NEUBIG; BERG-KIRKPATRICK, 2018)
- The Return of Lexical Dependencies: Neural Lexicalized PCFGs (ZHU; BISK; NEUBIG, 2020)
- Neural Bi-Lexicalized PCFG Induction (YANG; ZHAO; TU, 2021)
- StructFormer:Joint Unsupervised Induction of Dependency and Constituency Structure from Masked Language Modeling (SHEN *et al.*, 2021)

Destes trabalhos, nenhum foi testado em língua portuguesa, assim como todos eles testaram gramática de dependência a partir da conversão de gramática de constituintes. O estudo mais próximo testado na língua portuguesa utilizando gramática de dependência sem conversão foi implementado por Spitkovsky, Alshawi e Jurafsky (2013). No entanto, este estudo não utiliza texto cru para o seu treinamento. Portanto, para realizar uma comparação do modelo apresentado neste trabalho com um modelo generalista, o modelo DMV foi selecionado como *baseline*, uma vez que ele é amplamente utilizado em diferentes modelos. Os resultados demonstraram que o DMV aplicado à língua portuguesa teve um desempenho inferior tanto em UDA quanto em DDA, quando comparado ao modelo IM. O modelo apresentado neste trabalho foi, inclusive, superior a modelos de língua com *shot* um e *shot* zero.

Os resultados obtidos mostraram que até mesmo um método simples como a informação mútua pode proporcionar bons resultados, inclusive para idiomas em risco de extinção, como as línguas indígenas.

8.1 Contribuições

A primeira contribuição deste trabalho foi a apresentação de um mapeamento sistemático sobre a área dos últimos 20 anos. Ao contrário dos mapeamentos e revisões sistemáticas anteriores (SANKARAN, 2010; MURALIDARAN; SPASIĆ; KNIGHT, 2021; MARECEK, 2016; D'ULIZIA; FERRI; GRIFONI, 2011b), a abordagem apresentada neste estudo oferece uma análise mais abrangente e detalhada focada exclusivamente nos modelos de indução gramatical não supervisionada. A segunda contribuição foi a apresentação de um estudo demonstrando que é possível utilizar informação mútua para descobrir relações sintáticas na língua portuguesa. Isso permitiu uma melhor exploração da informação mútua na tarefa de indução gramatical. A terceira contribuição foi a extensão da segunda contribuição para um grande número de línguas que demonstrou que a informação mútua pode ser utilizada para recuperar relações sintáticas em diferentes línguas, e que o sistema de escrita pode impactar na indução gramatical a depender da modelagem utilizada. Por fim, a última contribuição foi o uso de heurística para melhorar a performance em UDA em língua portuguesa.

8.2 Trabalhos futuros

Este trabalho abre novas possibilidade de desenvolvimento de aplicações para a língua portuguesa que necessitam da indução gramatical, por exemplo, auxiliar na preservação de línguas ameaçadas de extinção. O presente método pode ser aprimorado a partir do uso de otimização de forma que seja possível aumentar o escopo de recuperação de relações sintáticas, onde neste trabalho tem o teto de 69%.

8.3 Publicações

Publicações	Relacionado com a dissertação	Situação
DA SILVA, D. P. G., & PARDO, T. A. S. **Induçao Gramatical para o Português: a Contribuição da Informaçao Mutua para Descoberta de Relações de Dependência**. In: Anais do XIV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana. SBC, 2023.	Sim	Publicado
DA SILVA, D. P. G., & PARDO, T. A. S. **Grammar Induction for Brazilian Indigenous Languages** . In: The 16th International Conference on Computational Processing of Portuguese, 2024.	Sim	Publicado
DA SILVA, D. P. G., & PARDO, T. A. S. **Using Mutual Information to discover dependency relations across 69 languages**.	Sim	Em análise
DA SILVA, D. P. G., & PARDO, T. A. S. **Unsupervised Grammar Induction in Natural Langauge Processing: A systematic mapping review**.	Sim	Em análise

REFERÊNCIAS

ABDULLAH, M.; MADAIN, A.; JARARWEH, Y. Chatgpt: Fundamentals, applications and social impacts. In: IEEE. **2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)**. [S.l.], 2022. p. 1–8. Citado na página 130.

AFONSO, S.; BICK, E.; HABER, R.; SANTOS, D. Floresta sintá (c) tica: a treebank for portuguese. In: ELRA. **quot; In Manuel González Rodrigues; Carmen Paz Suarez Araujo (ed) Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)(Las Palmas de Gran Canaria Espanha 29-31 de Maio de 2002) Paris: ELRA**. [S.l.], 2002. Citado nas páginas 46 e 47.

AHO, A. V.; SETHI, R.; ULLMAN, J. D. **Compilers: principles, techniques, and tools**. [S.l.]: Addison-wesley Reading, 2007. v. 2. Citado na página 32.

AKMAJIAN, A.; FARMER, A. K.; BICKMORE, L.; DEMERS, R. A.; HARNISH, R. M. **Linguistics: An introduction to language and communication**. [S.l.]: MIT press, 2017. Citado na página 32.

ALLERTON, D. J. Valency and the english verb. **(No Title)**, 1982. Citado na página 43.

ANDREW, e. a. Unsupervised latent tree induction with deep inside-outside recursive autoencoders. **NAACL-HLT**, ACL, v. 38, p. 453–468, 2019. Citado nas páginas 25, 29 e 38.

AULETE, F. J. d. C. Diccionario contemporaneo da lingua portugueza. Imprensa nacional, 1881. Citado na página 31.

BACKUS, J. W.; BAUER, F. L.; GREEN, J.; KATZ, C.; MCCARTHY, J.; PERLIS, A. J.; RUTISHAUSER, H.; SAMELSON, K.; VAUQUOIS, B.; WEGSTEIN, J. H. *et al.* Report on the algorithmic language algol 60. **Communications of the ACM**, ACM New York, NY, USA, v. 3, n. 5, p. 299–314, 1960. Citado na página 33.

BAKER, J. K. Trainable grammars for speech recognition. **The Journal of the Acoustical Society of America**, Acoustical Society of America, v. 65, n. S1, p. S132–S132, 1979. Citado na página 40.

BANNARD, C.; LIEVEN, E.; TOMASELLO, M. Modeling children's early grammatical knowledge. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 106, n. 41, p. 17284–17289, 2009. Citado na página 24.

BARAZANDEH, B.; RAZAVIYAYN, M. On the behavior of the expectation-maximization algorithm for mixture models. In: **2018 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2018, Anaheim, CA, USA, November 26-29, 2018**. IEEE, 2018. p. 61–65. Disponível em: <https://doi.org/10.1109/GlobalSIP.2018.8646506>. Citado na página 43.

BATISTA, V. F. L.; AGUILAR, R.; ALONSO, L.; GARCÍA, M. N. M. Data mining for grammatical inference with bioinformatics criteria. **Expert Syst. Appl.**, v. 39, n. 3, p. 2330–2334, 2012. Disponível em: <https://doi.org/10.1016/j.eswa.2011.08.058>. Citado na página 24.

BECHARA, E. **Moderna gramática portuguesa**. [S.l.]: Nova Fronteira, 2012. Citado na página 32.

BERWICK, R. C.; PIETROSKI, P.; YANKAMA, B.; CHOMSKY, N. Poverty of the stimulus revisited. **Cognitive Science**, Wiley Online Library, v. 35, n. 7, p. 1207–1242, 2011. Citado na página 32.

BLACK, E.; ABNEY, S.; FLICKINGER, D.; GDANIEC, C.; GRISHMAN, R.; HARRISON, P.; HINDLE, D.; INGRIA, R.; JELINEK, F.; KLAVANS, J. L. *et al.* A procedure for quantitatively comparing the syntactic coverage of english grammars. In: **Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991**. [S.l.: s.n.], 1991. Citado na página 52.

BLEI, D. M.; KUCUKELBIR, A.; MCAULIFFE, J. D. Variational inference: A review for statisticians. **Journal of the American statistical Association**, Taylor & Francis, v. 112, n. 518, p. 859–877, 2017. Citado nas páginas 40, 41 e 42.

BOD, R. An all-subtrees approach to unsupervised parsing. In: **Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2006. p. 865–872. Citado nas páginas 37, 39, 40 e 52.

____. Unsupervised parsing with U-DOP. In: MÀRQUEZ, L.; KLEIN, D. (Ed.). **Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL 2006, New York City, USA, June 8-9, 2006**. ACL, 2006. p. 85–92. Disponível em: <https://aclanthology.org/W06-2912/>. Citado na página 39.

____. Is the end of supervised parsing in sight? In: CARROLL, J.; BOSCH, A. van den; ZAENEN, A. (Ed.). **ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic**. The Association for Computational Linguistics, 2007. Disponível em: <https://aclanthology.org/P07-1051/>. Citado na página 37.

____. From exemplar to grammar: A probabilistic analogy-based model of language learning. **Cognitive Science**, Wiley Online Library, v. 33, n. 5, p. 752–793, 2009. Citado na página 24.

BRANCO, A.; SILVA, J.; GOMES, L.; RODRIGUES, J. Universal grammatical dependencies for portuguese with cintil data, lx processing and clarin support. In: **Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC), Marseille**. [S.l.: s.n.], 2022. p. 5617–5626. Citado na página 47.

BRESNAN, J.; ASUDEH, A.; TOIVONEN, I.; WECHSLER, S. **Lexical-functional syntax**. [S.l.]: John Wiley & Sons, 2015. Citado na página 35.

BROWN, T.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J. D.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A. *et al.* Language models are few-shot learners. **Advances in neural information processing systems**, v. 33, p. 1877–1901, 2020. Citado na página 130.

BUCHHOLZ, S.; MARSI, E. Conll-x shared task on multilingual dependency parsing. In: MÀRQUEZ, L.; KLEIN, D. (Ed.). **Proceedings of the Tenth Conference on Computational Natural Language Learning, CoNLL 2006, New York City, USA, June 8-9, 2006**. ACL, 2006. p. 149–164. Disponível em: <https://aclanthology.org/W06-2920/>. Citado nas páginas 45 e 49.

CAI, J.; JIANG, Y.; TU, K. CRF autoencoder for unsupervised dependency parsing. Association for Computational Linguistics, p. 1638–1643, 2017. Disponível em: <https://doi.org/10.18653/v1/d17-1171>. Citado nas páginas 28, 38 e 45.

CANCHO, R. F. i; SOLÉ, R. V.; KÖHLER, R. Patterns in syntactic dependency networks. **Physical Review E**, APS, v. 69, n. 5, p. 051915, 2004. Citado na página 36.

CARROLL, G.; CHARNIAK, E. **Two experiments on learning probabilistic dependency grammars from corpora**. [S.l.]: Department of Computer Science, Univ., 1992. Citado nas páginas 36 e 37.

CHEN, S. F. Bayesian grammar induction for language modeling. In: USZKOREIT, H. (Ed.). **33rd Annual Meeting of the Association for Computational Linguistics, 26-30 June 1995, MIT, Cambridge, Massachusetts, USA, Proceedings**. Morgan Kaufmann Publishers / ACL, 1995. p. 228–235. Disponível em: <https://aclanthology.org/P95-1031/>. Citado na página 44.

CHOMSKY, N. Three models for the description of language. **IRE Transactions on information theory**, IEEE, v. 2, n. 3, p. 113–124, 1956. Citado nas páginas 31, 32 e 33.

____. Syntactic structures. 1957. Citado na página 35.

____. On certain formal properties of grammars. **Information and control**, Elsevier, v. 2, n. 2, p. 137–167, 1959. Citado na página 35.

____. **On nature and language**. [S.l.]: Cambridge University Press, 2002. Citado na página 32.

____. **Aspects of the Theory of Syntax**. [S.l.]: MIT press, 2014. v. 11. Citado nas páginas 31, 32 e 33.

CHURCH, K. W.; HANKS, P. Word association norms, mutual information, and lexicography. **Comput. Linguistics**, v. 16, n. 1, p. 22–29, 1990. Citado na página 29.

CLARK, A. Unsupervised induction of stochastic context-free grammars using distributional clustering. In: DAELEMANS, W.; ZAJAC, R. (Ed.). **Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning, CoNLL 2001, Toulouse, France, July 6-7, 2001**. ACL, 2001. Disponível em: <https://aclanthology.org/W01-0713/>. Citado nas páginas 44 e 45.

____. Grammatical inference and first language acquisition. In: **Proceedings of the Workshop on Psycho-Computational Models of Human Language Acquisition**. [S.l.: s.n.], 2004. p. 27–34. Citado na página 24.

COHEN, M.; CACIULARU, A.; REJWAN, I.; BERANT, J. Inducing regular grammars using recurrent neural networks. **CoRR**, abs/1710.10453, 2017. Disponível em: <http://arxiv.org/abs/1710.10453>. Citado na página 24.

COHEN, S. B.; GIMPEL, K.; SMITH, N. A. Logistic normal priors for unsupervised probabilistic grammar induction. In: KOLLER, D.; SCHUURMANS, D.; BENGIO, Y.; BOTTOU, L. (Ed.). **Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008**. Curran Associates, Inc., 2008. p. 321–328. Disponível em: <https://proceedings.neurips.cc/paper/2008/hash/f11bec1411101c743f64df596773d0b2-Abstract.html>. Citado nas páginas 41 e 128.

COHEN, S. B.; SMITH, N. A. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In: **Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA**. The Association for Computational Linguistics, 2009. p. 74–82. Disponível em: <https://aclanthology.org/N09-1009/>. Citado na página 25.

COHN, T.; BLUNSOM, P.; GOLDWATER, S. Inducing tree-substitution grammars. **The Journal of Machine Learning Research**, JMLR. org, v. 11, p. 3053–3096, 2010. Citado na página 40.

COLLINS, M. Head-driven statistical models for natural language parsing. **Comput. Linguistics**, v. 29, n. 4, p. 589–637, 2003. Disponível em: <https://doi.org/10.1162/089120103322753356>. Citado na página 48.

COURTIN, J.; GENTHIAL, D. Parsing with dependency relations and robust parsing. In: **Processing of Dependency-Based Grammars**. [S.l.: s.n.], 1998. Citado na página 26.

CSISZÁR, I. I-divergence geometry of probability distributions and minimization problems. **The annals of probability**, JSTOR, p. 146–158, 1975. Citado na página 41.

DAHL, V.; BEL-ENGUIX, G.; TIRADO, V.; MIRALLES, E. Grammar induction for under-resourced languages: The case of ch'ol. In: **Analysis, Verification and Transformation for Declarative Programming and Intelligent Systems: Essays Dedicated to Manuel Hermenegildo on the Occasion of His 60th Birthday**. [S.l.]: Springer, 2023. p. 113–132. Citado na página 24.

DALE, R.; MOISL, H.; SOMERS, H. Handbook of natural language processing. **Imprint New York: Marcel Dekker**, 2000. Citado na página 31.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. **Journal of the royal statistical society: series B (methodological)**, Wiley Online Library, v. 39, n. 1, p. 1–22, 1977. Citado na página 40.

D'ULIZIA, A.; FERRI, F.; GRIFONI, P. A survey of grammatical inference methods for natural language learning. **Artif. Intell. Rev.**, v. 36, n. 1, p. 1–27, 2011. Disponível em: <https://doi.org/10.1007/s10462-010-9199-1>. Citado na página 52.

____. A survey of grammatical inference methods for natural language learning. **Artif. Intell. Rev.**, v. 36, n. 1, p. 1–27, 2011. Disponível em: <https://doi.org/10.1007/s10462-010-9199-1>. Citado na página 140.

DYRKA, W.; GĄSIOR-GŁOGOWSKA, M.; SZEFCZYK, M.; SZULC, N. Searching for universal model of amyloid signaling motifs using probabilistic context-free grammars. **BMC bioinformatics**, Springer, v. 22, n. 1, p. 222, 2021. Citado na página 24.

EL-SHISHINY, H. A formal description of arabic syntax in definite clause grammar. In: **COLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics**. [S.l.: s.n.], 1990. Citado na página 35.

EVANS, V. **Cognitive linguistics**. [S.l.]: Edinburgh University Press, 2006. Citado na página 32.

FUTRELL, R.; QIAN, P.; GIBSON, E.; FEDORENKO, E.; BLANK, I. Syntactic dependencies correspond to word pairs with high mutual information. In: **Proceedings of the fifth international conference on dependency linguistics (depling, syntaxfest 2019)**. [S.l.: s.n.], 2019. p. 3–13. Citado nas páginas 29, 30 e 91.

GERSHMAN, S. J.; BLEI, D. M. A tutorial on bayesian nonparametric models. **Journal of Mathematical Psychology**, Elsevier, v. 56, n. 1, p. 1–12, 2012. Citado na página 40.

GILLENWATER, J.; GANCHEV, K.; GRAÇA, J.; PEREIRA, F.; TASKAR, B. Posterior sparsity in unsupervised dependency parsing. **The Journal of Machine Learning Research**, JMLR. org, v. 12, p. 455–490, 2011. Citado na página 43.

GOLD, E. M. Language identification in the limit. **Information and control**, Elsevier, v. 10, n. 5, p. 447–474, 1967. Citado na página 27.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016. Citado na página 44.

GOODMAN, J. C.; DALE, P. S.; LI, P. Does frequency count? parental input and the acquisition of vocabulary. **Journal of child language**, Cambridge University Press, v. 35, n. 3, p. 515–531, 2008. Citado na página 126.

GRAVE, E.; ELHADAD, N. A convex and feature-rich discriminative approach to dependency grammar induction. In: **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**. [S.l.: s.n.], 2015. p. 1375–1384. Citado nas páginas 27 e 28.

GUIBON, G.; COURTIN, M.; GERDES, K.; GUILLAUME, B. When collaborative treebank curation meets graph grammars. In: **Proceedings of The 12th Language Resources and Evaluation Conference**. Marseille, France: European Language Resources Association, 2020. p. 5293–5302. Disponível em: <https://www.aclweb.org/anthology/2020.lrec-1.651>. Citado na página 27.

HAJIČ, J. Building a syntactically annotated corpus: The prague dependency treebank. **Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová**, p. 106–132, 1998. Citado na página 48.

HAN, W.; JIANG, Y.; TU, K. Dependency grammar induction with neural lexicalization and big training data. In: PALMER, M.; HWA, R.; RIEDEL, S. (Ed.). **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017**. Association for Computational Linguistics, 2017. p. 1683–1688. Disponível em: <https://doi.org/10.18653/v1/d17-1176>. Citado nas páginas 29 e 44.

____. Enhancing unsupervised generative dependency parser with contextual information. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2019. p. 5315–5325. Citado nas páginas 28, 29 e 38.

HARRIS, R. A. **The linguistics wars: Chomsky, Lakoff, and the battle over deep structure**. [S.l.]: Oxford University Press, 2021. Citado nas páginas 31, 32 e 35.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; RODRIGUES, J.; ALUISIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. **In Proceedings of Symposium in Information and Human Language Technology (STIL), Uberlândia**, 2017. Citado na página 129.

HAYS, D. G. Dependency theory: A formalism and some observations. **Language**, JSTOR, v. 40, n. 4, p. 511–525, 1964. Citado nas páginas 26 e 37.

HE, J.; NEUBIG, G.; BERG-KIRKPATRICK, T. Unsupervised learning of syntactic structure with invertible neural projections. In: RILOFF, E.; CHIANG, D.; HOCKENMAIER, J.; TSUJII, J. (Ed.). **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018**. Association for Computational Linguistics, 2018. p. 1292–1302. Disponível em: <https://doi.org/10.18653/v1/d18-1160>. Citado nas páginas 30, 129 e 140.

HIGUERA, C. D. L. A bibliographical study of grammatical inference. **Pattern recognition**, Elsevier, v. 38, n. 9, p. 1332–1348, 2005. Citado nas páginas 24 e 25.

HIRSCHBERG, J.; MANNING, C. D. Advances in natural language processing. **Science**, American Association for the Advancement of Science, v. 349, n. 6245, p. 261–266, 2015. Citado na página 23.

HOOVER, J. L.; DU, W.; SORDONI, A.; O'DONNELL, T. J. Linguistic dependencies and statistical dependence. Association for Computational Linguistics, p. 2941–2963, 2021. Disponível em: <https://doi.org/10.18653/v1/2021.emnlp-main.234>. Citado na página 29.

HOVY, E.; LAVID, J. Towards a 'science'of corpus annotation: a new methodological challenge for corpus linguistics. **International journal of translation**, v. 22, n. 1, p. 13–36, 2010. Citado na página 24.

III, W. P. H.; JOHNSON, M.; MCCLOSKY, D. Improving unsupervised dependency parsing with richer contexts and smoothing. In: **Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the association for computational linguistics**. [S.l.: s.n.], 2009. p. 101–109. Citado na página 38.

JÄGER, G.; ROGERS, J. Formal language theory: refining the chomsky hierarchy. **Philosophical Transactions of the Royal Society B: Biological Sciences**, The Royal Society, v. 367, n. 1598, p. 1956–1970, 2012. Citado na página 32.

JAWAHAR, G.; SAGOT, B.; SEDDAH, D. What does bert learn about the structure of language? In: **ACL 2019-57th Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2019. Citado na página 24.

JIANG, Y.; HAN, W.; TU, K. Unsupervised neural dependency parsing. In: **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2016. p. 763–771. Citado nas páginas 29 e 43.

JIN, L.; DOSHI-VELEZ, F.; MILLER, T. A.; SCHULER, W.; SCHWARTZ, L. Unsupervised grammar induction with depth-bounded PCFG. **Trans. Assoc. Comput. Linguistics**, v. 6, p. 211–224, 2018. Disponível em: <https://doi.org/10.1162/tacl_a_00016>. Citado nas páginas 25 e 41.

JIN, L.; DOSHI-VELEZ, F.; MILLER, T. A.; SCHWARTZ, L.; SCHULER, W. Unsupervised learning of pcfgs with normalizing flow. In: KORHONEN, A.; TRAUM, D. R.; MÀRQUEZ, L. (Ed.). **Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers**. Association for Computational Linguistics, 2019. p. 2442–2452. Disponível em: <https://doi.org/10.18653/v1/p19-1234>. Citado na página 41.

JURAFSKY, D. **Speech & language processing**. [S.l.]: Pearson Education India, 2000. Citado nas páginas 25, 33 e 36.

JW, B. c the syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference. In: **Conference, on,, Inform&on, Processing**. [S.l.: s.n.], 1959. p. 125–131. Citado na página 33.

KIM, T.; CHOI, J.; EDMISTON, D.; LEE, S. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. OpenReview.net, 2020. Disponível em: <https://openreview.net/forum?id=H1xPR3NtPB>. Citado na página 44.

KLEIN, D. **The unsupervised learning of natural language structure**. [S.l.]: Stanford University, 2005. Citado na página 52.

KLEIN, D.; MANNING, C. D. A generative constituent-context model for improved grammar induction. In: **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA**. ACL, 2002. p. 128–135. Disponível em: <https://aclanthology.org/P02-1017/>. Citado nas páginas 25, 27, 28, 38, 39 e 52.

____. Corpus-based induction of syntactic structure: Models of dependency and constituency. In: SCOTT, D.; DAELEMANS, W.; WALKER, M. A. (Ed.). **Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain**. ACL, 2004. p. 478–485. Disponível em: <https://aclanthology.org/P04-1061/>. Citado nas páginas 15, 24, 27, 30, 36, 37, 44, 48 e 52.

KUHLMANN, M. **Dependency Structures and Lexicalized Grammars An Algebraic Approach**. Springer, 2010. v. 6270. (Lecture Notes in Computer Science, v. 6270). ISBN 978-3-642-14567-4. Disponível em: <https://doi.org/10.1007/978-3-642-14568-1>. Citado na página 36.

KULMIZEV, A.; NIVRE, J. Schrödinger's tree—on syntax and neural language models. **Frontiers in Artificial Intelligence**, Frontiers, v. 5, p. 796788, 2022. Citado nas páginas 23 e 24.

LAKOFF, G. Woman, fire, and dangerous things. **What Categories Reveal about the Mind**, University of Chicago Press, 1987. Citado na página 32.

LANGACKER, R. W. **Foundations of cognitive grammar: Volume I: Theoretical prerequisites**. [S.l.]: Stanford university press, 1987. v. 1. Citado nas páginas 31 e 32.

____. Structural syntax: the view from cognitive grammar. **Dondelinger et al.[1995]**, p. 13–37, 1995. Citado na página 35.

LASKAR, M. T. R.; BARI, M. S.; RAHMAN, M.; BHUIYAN, M. A. H.; JOTY, S.; HUANG, J. X. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. In: ROGERS, A.; BOYD-GRABER, J. L.; OKAZAKI, N. (Ed.). **Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023**. Association for

Computational Linguistics, 2023. p. 431–469. Disponível em: <https://doi.org/10.18653/v1/2023.findings-acl.29>. Citado nas páginas 23 e 130.

LI, B.; CHENG, J.; LIU, Y.; KELLER, F. Dependency grammar induction with a neural variational transition-based parser. In: **The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019**. AAAI Press, 2019. p. 6658–6665. Disponível em: <https://doi.org/10.1609/aaai.v33i01.33016658>. Citado nas páginas 28 e 45.

LI, B.; CORONA, R.; MANGALAM, K.; CHEN, C.; FLAHERTY, D.; BELONGIE, S. J.; WEINBERGER, K. Q.; MALIK, J.; DARRELL, T.; KLEIN, D. Does unsupervised grammar induction need pixels? **CoRR**, abs/2212.10564, 2022. Disponível em: <https://doi.org/10.48550/arXiv.2212.10564>. Citado na página 24.

LIN, B.; YAO, Z.; SHI, J.; CAO, S.; TANG, B.; LI, S.; LUO, Y.; LI, J.; HOU, L. Dependency parsing via sequence generation. In: **Findings of the Association for Computational Linguistics (EMNLP),Abu Dhabi**. [S.l.: s.n.], 2022. p. 7339–7353. Citado na página 37.

LINGUATECA. Cetempúblico: Um corpus de grandes dimensões de linguagem jornalística portuguesa. **Linguateca,http://www.linguateca.pt/CETENFolha/, última visita: Junho de 2023**, ICMC/USP, 2023. Citado na página 47.

MAGERMAN, D. M.; MARCUS, M. P. Parsing a natural language using mutual information statistics. In: SHROBE, H. E.; DIETTERICH, T. G.; SWARTOUT, W. R. (Ed.). **Proceedings of the 8th National Conference on Artificial Intelligence. Boston, Massachusetts, USA, July 29 - August 3, 1990, 2 Volumes**. AAAI Press / The MIT Press, 1990. p. 984–989. Disponível em: <http://www.aaai.org/Library/AAAI/1990/aaai90-147.php>. Citado na página 29.

MARCUS, M. P.; SANTORINI, B.; MARCINKIEWICZ, M. A. Building a large annotated corpus of english: The penn treebank. **Comput. Linguistics**, v. 19, n. 2, p. 313–330, 1993. Citado nas páginas 46 e 48.

MARECEK, D. Twelve years of unsupervised dependency parsing. In: **ITAT**. [S.l.: s.n.], 2016. p. 56–62. Citado na página 140.

MAREČEK, D.; ŽABOKRTSKÝ, Z. Gibbs sampling with treeness constraint in unsupervised dependency parsing. In: **Proceedings of Workshop on Robust Unsupervised and Semisupervised Methods in Natural Language Processing**. [S.l.: s.n.], 2011. p. 1–8. Citado nas páginas 40 e 52.

MARNEFFE, M.-C. D.; DOZAT, T.; SILVEIRA, N.; HAVERINEN, K.; GINTER, F.; NIVRE, J.; MANNING, C. D. Universal stanford dependencies: A cross-linguistic typology. In: **LREC**. [S.l.: s.n.], 2014. v. 14, p. 4585–4592. Citado nas páginas 15, 36, 37 e 50.

MARNEFFE, M.-C. D.; MANNING, C. D.; NIVRE, J.; ZEMAN, D. Universal dependencies. **Computational linguistics**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 47, n. 2, p. 255–308, 2021. Citado nas páginas 27, 30, 36 e 50.

MARNEFFE, M.-C. D.; NIVRE, J. Dependency grammar. **Annual Review of Linguistics**, Annual Reviews, v. 5, p. 197–218, 2019. Citado nas páginas 26, 35 e 36.

MATTHAIOS, S. Eratosthenes of cyrene: Readings of his 'grammar'definition. **Ancient Scholarship and Grammar: Archetypes, Concepts and Contexts. Trends in classics-supplementary volumes**, v. 8, p. 55–86, 2011. Citado na página 31.

MAVELI, N.; COHEN, S. B. Co-training an unsupervised constituency parser with weak supervision. Association for Computational Linguistics, p. 1274–1291, 2022. Disponível em: <https://doi.org/10.18653/v1/2022.findings-acl.101>. Citado na página 38.

MCDONALD, R. T.; NIVRE, J.; QUIRMBACH-BRUNDAGE, Y.; GOLDBERG, Y.; DAS, D.; GANCHEV, K.; HALL, K. B.; PETROV, S.; ZHANG, H.; TÄCKSTRÖM, O.; BEDINI, C.; CASTELLÓ, N. B.; LEE, J. Universal dependency annotation for multilingual parsing. In: **Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers**. The Association for Computer Linguistics, 2013. p. 92–97. Disponível em: <https://aclanthology.org/P13-2017/>. Citado na página 49.

MCENERY, T. **Corpus linguistics**. [S.l.]: Edinburgh University Press, 2019. Citado na página 24.

MCLACHLAN, G. J.; KRISHNAN, T. **The EM algorithm and extensions**. [S.l.]: John Wiley & Sons, 2007. Citado na página 42.

MEL'CUK, I. A. *et al.* **Dependency syntax: theory and practice**. [S.l.]: SUNY press, 1988. Citado na página 36.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. **Advances in neural information processing systems**, v. 26, 2013. Citado nas páginas 124 e 129.

MITKOV, R. **The Oxford handbook of computational linguistics**. [S.l.]: Oxford University Press, 2022. Citado nas páginas 32 e 36.

MURALIDARAN, V.; SPASIĆ, I.; KNIGHT, D. A systematic review of unsupervised approaches to grammar induction. **Natural Language Engineering**, Cambridge University Press, v. 27, n. 6, p. 647–689, 2021. Citado na página 140.

MURPHY, K. P. **Probabilistic machine learning: an introduction**. [S.l.]: MIT press, 2022. Citado nas páginas 39, 40 e 41.

NIVRE, J. Towards a universal grammar for natural language processing. In: GELBUKH, A. F. (Ed.). **Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part I**. Springer, 2015. (Lecture Notes in Computer Science, v. 9041), p. 3–16. Disponível em: <https://doi.org/10.1007/978-3-319-18111-0_1>. Citado na página 49.

NIVRE, J.; MARNEFFE, M. de; GINTER, F.; HAJIC, J.; MANNING, C. D.; PYYSALO, S.; SCHUSTER, S.; TYERS, F. M.; ZEMAN, D. Universal dependencies v2: An evergrowing multilingual treebank collection. In: CALZOLARI, N.; BÉCHET, F.; BLACHE, P.; CHOUKRI, K.; CIERI, C.; DECLERCK, T.; GOGGI, S.; ISAHARA, H.; MAEGAARD, B.; MARIANI, J.; MAZO, H.; MORENO, A.; ODIJK, J.; PIPERIDIS, S. (Ed.). **Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020**. European Language Resources Association, 2020. p. 4034–4043. Disponível em: <https://aclanthology.org/2020.lrec-1.497/>. Citado na página 49.

PAGANI, L. A. Diagramas em árvore como representação da estrutura sintática. **UFPR,https://docs.ufpr.br/∼arthur/textos/apr/sintaxe/arv_apr.pdf, última visita: junho de 2023**, 2018. Citado na página 25.

PAISLEY, J.; WANG, C.; BLEI, D. The discrete infinite logistic normal distribution for mixed-membership modeling. In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. **Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics**. [S.l.], 2011. p. 74–82. Citado na página 41.

PARDO, T. A. S.; DURAN, M. S.; LOPES, L.; FELIPPO, A. d.; ROMAN, N. T.; NUNES, M. d. G. V. Porttinari: a large multi-genre treebank for brazilian portuguese. **Anais**, 2021. Citado nas páginas 47 e 124.

PARIKH, A.; COHEN, S. B.; XING, E. Spectral unsupervised parsing with additive tree metrics. In: **Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. [S.l.: s.n.], 2014. p. 1062–1072. Citado nas páginas 38 e 39.

PATE, J. K.; JOHNSON, M. Grammar induction from (lots of) words alone. In: CALZOLARI, N.; MATSUMOTO, Y.; PRASAD, R. (Ed.). **COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan**. ACL, 2016. p. 23–32. Disponível em: <https://aclanthology.org/C16-1003/>. Citado na página 52.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL**. ACL, 2014. p. 1532–1543. Disponível em: <https://doi.org/10.3115/v1/d14-1162>. Citado nas páginas 124 e 129.

PETROV, S.; DAS, D.; MCDONALD, R. T. A universal part-of-speech tagset. In: CALZOLARI, N.; CHOUKRI, K.; DECLERCK, T.; DOGAN, M. U.; MAEGAARD, B.; MARIANI, J.; ODIJK, J.; PIPERIDIS, S. (Ed.). **Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012**. European Language Resources Association (ELRA), 2012. p. 2089–2096. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2012/summaries/274.html>. Citado na página 49.

PIRES, R.; ABONIZIO, H.; ROGÉRIO, T.; NOGUEIRA, R. Sabi\'a: Portuguese large language models. **arXiv preprint arXiv:2304.07880**, 2023. Citado na página 130.

RADEMAKER, A.; CHALUB, F.; REAL, L.; FREITAS, C.; BICK, E.; PAIVA, V. D. Universal dependencies for portuguese. In: **Proceedings of the fourth international conference on dependency linguistics (Depling 2017)**. [S.l.: s.n.], 2017. p. 197–206. Citado na página 47.

ROCHA, P. A.; SANTOS, D. Cetempúblico: Um corpus de grandes dimensões de linguagem jornalística portuguesa. **quot; In Maria das Graças Volpe Nunes (ed) V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR 2000)(Atibaia SP 19-22 de Novembro de 2000) São Paulo: ICMC/USP**, ICMC/USP, 2000. Citado na página 47.

ROGERS, S.; GIROLAMI, M. **A first course in machine learning**. [S.l.]: Chapman and Hall/CRC, 2016. Citado na página 41.

ROTMAN, G.; REICHART, R. Deep contextualized self-training for low resource dependency parsing. **Transactions of the Association for Computational Linguistics**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 7, p. 695–713, 2019. Citado na página 38.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *et al.* **Learning internal representations by error propagation**. [S.l.]: Institute for Cognitive Science, University of California, San Diego La . . . , 1985. Citado na página 43.

SAG, I. A.; BALDWIN, T.; BOND, F.; COPESTAKE, A.; FLICKINGER, D. Multiword expressions: A pain in the neck for nlp. In: SPRINGER. **Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings 3**. [S.l.], 2002. p. 1–15. Citado na página 51.

SANKARAN, B. A survey of unsupervised grammar induction. **Manuscript, Simon Fraser University**, Citeseer, v. 47, 2010. Citado nas páginas 40, 52, 128 e 140.

SANTAMARIA, J.; ARAUJO, L. Identifying patterns for unsupervised grammar induction. In: **Proceedings of the Fourteenth Conference on Computational Natural Language Learning**. [S.l.: s.n.], 2010. p. 38–45. Citado na página 123.

SARDINHA, T. B. **Lingüística de corpus**. [S.l.]: Editora Manole Ltda, 2004. Citado na página 24.

SCHANK, R. C.; TESLER, L. A conceptual dependency parser for natural language. In: **International Conference on Computational Linguistics COLING 1969: Preprint No. 2**. [S.l.: s.n.], 1969. Citado na página 36.

SCIULLO, A. M. D.; PIATTELLI-PALMARINI, M.; WEXLER, K.; BERWICK, R. C.; BOECKX, C.; JENKINS, L.; URIAGEREKA, J.; STROMSWOLD, K.; CHENG, L. L.-S.; HARLEY, H. *et al.* The biological nature of human language. **Biolinguistics**, v. 4, n. 1, p. 004–034, 2010. Citado na página 32.

SEGINER, Y. Fast unsupervised incremental parsing. In: CARROLL, J.; BOSCH, A. van den; ZAENEN, A. (Ed.). **ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic**. The Association for Computational Linguistics, 2007. Disponível em: <https://aclanthology.org/P07-1049/>. Citado nas páginas 39 e 40.

SEKINE, S.; COLLINS, M. Evalb bracket scoring program. **URL: http://www. cs. nyu. edu/cs/-projects/proteus/evalb**, 1997. Citado na página 52.

SHEN, Y.; LIN, Z.; HUANG, C.; COURVILLE, A. C. Neural language modeling by jointly learning syntax and lexicon. In: **6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings**. OpenReview.net, 2018. Disponível em: <https://openreview.net/forum?id=rkgOLb-0W>. Citado na página 43.

SHEN, Y.; TAY, Y.; ZHENG, C.; BAHRI, D.; METZLER, D.; COURVILLE, A. C. Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In: ZONG, C.; XIA, F.; LI, W.; NAVIGLI, R. (Ed.). **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers),**

Virtual Event, August 1-6, 2021. [S.l.]: Association for Computational Linguistics, 2021. p. 7196–7209. Citado nas páginas 29 e 140.

SHI, H.; MAO, J.; GIMPEL, K.; LIVESCU, K. Visually grounded neural syntax acquisition. In: KORHONEN, A.; TRAUM, D. R.; MÀRQUEZ, L. (Ed.). **Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers**. Association for Computational Linguistics, 2019. p. 1842–1861. Disponível em: <https://doi.org/10.18653/v1/p19-1180>. Citado na página 24.

SHIBATA, C. Learning (k, l)-context-sensitive probabilistic grammars with nonparametric bayesian approach. **Machine Learning**, Springer, p. 1–35, 2021. Citado na página 24.

SIDDHARTHAN, A. A survey of research on text simplification. **ITL-International Journal of Applied Linguistics**, John Benjamins, v. 165, n. 2, p. 259–298, 2014. Citado na página 24.

SIMA'AN, K. Computational complexity of probabilistic disambiguation by means of tree-grammars. In: **16th International Conference on Computational Linguistics, Proceedings of the Conference, COLING 1996, Center for Sprogteknologi, Copenhagen, Denmark, August 5-9, 1996**. [s.n.], 1996. p. 1175–1180. Disponível em: <https://aclanthology.org/C96-2215/>. Citado na página 36.

SIPSER, M. Introduction to the theory of computation. 2021. Citado nas páginas 33 e 35.

SKUT, W.; BRANTS, T.; KRENN, B.; USZKOREIT, H. A linguistically interpreted corpus of german newspaper text. **arXiv preprint cmp-lg/9807008**, 1998. Citado na página 48.

SMITH, N. A.; EISNER, J. Guiding unsupervised grammar induction using contrastive estimation. In: **Proc. of IJCAI Workshop on Grammatical Inference Applications**. [S.l.: s.n.], 2005. p. 73–82. Citado na página 48.

____. Annealing structural bias in multilingual weighted grammar induction. In: **Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics**. [S.l.: s.n.], 2006. p. 569–576. Citado na página 52.

SOLAN, Z.; HORN, D.; RUPPIN, E.; EDELMAN, S. Unsupervised context sensitive language acquisition from a large corpus. **Advances in Neural Information Processing Systems**, v. 16, 2003. Citado na página 52.

SOUZA, E. de; SILVEIRA, A.; CAVALCANTI, T.; CASTRO, M. C.; FREITAS, C. Petrogold-corpus padrão ouro para o domínio do petróleo. In: SBC. **Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (STIL), Online**. [S.l.], 2021. p. 29–38. Citado na página 47.

SPITKOVSKY, V. I.; ALSHAWI, H.; JURAFSKY, D. From baby steps to leapfrog: How "less is more"in unsupervised dependency parsing. In: **Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA**. The Association for Computational Linguistics, 2010. p. 751–759. Disponível em: <https://aclanthology.org/N10-1116/>. Citado nas páginas 36, 41 e 48.

____. Lateen EM: unsupervised training with multiple objectives, applied to dependency grammar induction. In: **Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL**. ACL, 2011. p. 1269–1280. Disponível em: <https://aclanthology.org/D11-1117/>. Citado na página 43.

____. Punctuation: Making a point in unsupervised dependency parsing. In: **Proceedings of the Fifteenth Conference on Computational Natural Language Learning**. [S.l.: s.n.], 2011. p. 19–28. Citado na página 123.

____. Capitalization cues improve dependency grammar induction. In: **Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure**. [S.l.: s.n.], 2012. p. 16–22. Citado na página 123.

____. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In: **Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL**. ACL, 2013. p. 1983–1995. Disponível em: <https://aclanthology.org/D13-1204/>. Citado na página 140.

SPITKOVSKY, V. I.; ALSHAWI, H.; JURAFSKY, D.; MANNING, C. D. Viterbi training improves unsupervised dependency parsing. ACL, p. 9–17, 2010. Disponível em: <https://aclanthology.org/W10-2902/>. Citado na página 43.

STAP, D.; ARAABI, A. Chatgpt is not a good indigenous translator. In: **Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)**. [S.l.: s.n.], 2023. p. 163–167. Citado na página 23.

STEEDMAN, M.; BALDRIDGE, J. Combinatory categorial grammar. **Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell**, p. 181–224, 2011. Citado nas páginas 33 e 35.

STEVENSON, A.; CORDY, J. R. A survey of grammatical inference in software engineering. **Sci. Comput. Program.**, v. 96, p. 444–459, 2014. Citado na página 24.

STOKEL-WALKER, C.; NOORDEN, R. V. What chatgpt and generative ai mean for science. **Nature**, Nature, v. 614, n. 7947, p. 214–216, 2023. Citado na página 23.

STOLCKE, A.; OMOHUNDRO, S. Inducing probabilistic grammars by bayesian model merging. In: SPRINGER. **International Colloquium on Grammatical Inference**. [S.l.], 1994. p. 106–118. Citado na página 39.

SUEN, C. Y. n-gram statistics for natural language understanding and text processing. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 1, n. 2, p. 164–172, 1979. Disponível em: <https://doi.org/10.1109/TPAMI.1979.4766902>. Citado na página 43.

TESNIÈRE, L. Éléments de syntaxe structurale. Paris, 1959. Citado na página 35.

TOMASELLO, M. Do young children have adult syntactic competence? **Cognition**, Elsevier, v. 74, n. 3, p. 209–253, 2000. Citado na página 32.

TOUVRON, H.; LAVRIL, T.; IZACARD, G.; MARTINET, X.; LACHAUX, M.-A.; LACROIX, T.; ROZIÈRE, B.; GOYAL, N.; HAMBRO, E.; AZHAR, F. *et al.* Llama: Open and efficient foundation language models. **arXiv preprint arXiv:2302.13971**, 2023. Citado na página 130.

TU, K.; JIANG, Y.; HAN, W.; ZHAO, Y. Unsupervised natural language parsing (introductory tutorial). In: **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts**. [S.l.: s.n.], 2021. p. 1–5. Citado na página 39.

UNOLD, O.; GABOR, M.; DYRKA, W. Unsupervised grammar induction for revealing the internal structure of protein sequence motifs. In: SPRINGER. **Artificial Intelligence in Medicine: 18th International Conference on Artificial Intelligence in Medicine, AIME 2020, Minneapolis, MN, USA, August 25–28, 2020, Proceedings 18**. [S.l.], 2020. p. 299–309. Citado na página 24.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017. Citado na página 44.

WANG, H.; WU, H.; HE, Z.; HUANG, L.; CHURCH, K. W. Progress in machine translation. **Engineering**, Elsevier, 2021. Citado na página 23.

WINTNER, S. Computational models of language acquisition. In: SPRINGER. **Computational Linguistics and Intelligent Text Processing: 11th International Conference, CICLing 2010, Iaşi, Romania, March 21-27, 2010. Proceedings 11**. [S.l.], 2010. p. 86–99. Citado na página 24.

XUE, N.; XIA, F.; CHIOU, F.-D.; PALMER, M. The penn chinese treebank: Phrase structure annotation of a large corpus. **Natural language engineering**, Cambridge University Press, v. 11, n. 2, p. 207–238, 2005. Citado na página 48.

YANG, S.; JIANG, Y.; HAN, W.; TU, K. Second-order unsupervised neural dependency parsing. In: SCOTT, D.; BEL, N.; ZONG, C. (Ed.). **Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020**. [S.l.]: International Committee on Computational Linguistics, 2020. p. 3911–3924. Citado nas páginas 29 e 38.

YANG, S.; ZHAO, Y.; TU, K. Neural bi-lexicalized PCFG induction. Association for Computational Linguistics, p. 2688–2699, 2021. Citado nas páginas 129 e 140.

ZEMAN, D.; HAJIC, J.; POPEL, M.; POTTHAST, M.; STRAKA, M.; GINTER, F.; NIVRE, J.; PETROV, S. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In: **Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies**. [S.l.: s.n.], 2018. p. 1–21. Citado na página 47.

ZHU, H.; BISK, Y.; NEUBIG, G. The return of lexical dependencies: Neural lexicalized pcfgs. **Transactions of the Association for Computational Linguistics**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 8, p. 647–661, 2020. Citado nas páginas 37, 43 e 140.

ZIPF, G. K. **Zipf, George Kingsley. "Human behavior and the principle of least effort."(1949).** [S.l.]: American Psychological Association, 1949. Citado nas páginas 124, 125 e 126.


ICMC
USP
SÃO CARLOS