Universidade de São Paulo - USP

Universidade Federal de São Carlos - UFSCar

Universidade Estadual Paulista - UNESP

SELEÇÃO E PREPARAÇÃO DE SENTENÇAS DO CORPUS PLN-BR PARA COMPOR O CORPUS DE ANOTAÇÃO DE PAPÉIS SEMÂNTICOS PROPBANK-BR.V2

Magali Sanches Duran

Lianet Sepúlveda-Torres

Marina Coimbra Viviani

Nathan Siegle Hartmann

Sandra Maria Aluísio

NILC-TR-14-07

Agosto, 2014

Série de Relatórios do Núcleo Interinstitucional de Linguística Computacional

NILC - ICMC-USP, Caixa Postal 668, 13560-970 São Carlos, SP, Brasil

SELEÇÃO E PREPARAÇÃO DE SENTENÇAS DO CORPUS PLN-BR PARA COMPOR O CORPUS DE ANOTAÇÃO DE PAPÉIS SEMÂNTICOS PROPBANK-BR.V2

SUMÁRIO

1. INTRODUÇÃO

O projeto Propbank-Br (Duran e Aluísio, 2012, 2011) adicionou uma camada de anotação de papéis semânticos à porção brasileira do corpus Bosque (Afonso et. al, 2002), que é um *treebank*, ou seja, um corpus anotado sintaticamente por um parser automático e corrigido manualmente por linguistas.

Cada uma das 4213 sentenças da porção brasileira do *treebank* original foi clonada tantas vezes quanto fosse o número de verbos plenos presentes nela. Cada cópia da sentença é chamada de "instância" e dedicada à anotação de um dos verbos plenos que ocorrem na sentença. Os verbos auxiliares não foram considerados evocadores de estrutura argumental (foco de anotação), pois eles são tratados como modificadores da estrutura dos verbos plenos. O processo gerou 7107 instâncias de anotação com 1068 verbos alvo de anotação.

O verbo "ser" era responsável por 965 instâncias (da 5270 à 6234) e, como ele tem um único sentido (verbo de cópula) não foi anotado na primeira versão do Propbank-Br (na versão 1.1, 200 instâncias desse verbo foram anotadas). No final, o corpus ficou com 6142 instâncias anotadas e foi usado para treinamento de dois classificadores de papéis semânticos (Alva-Manchego, 2013; Fonseca, 2013). As 306 instâncias que apresentavam erros ou de análise sintática ou de corpus (erros de ortografia, por exemplo), foram anotadas com a etiqueta "wrongsubcorpus". Essas instâncias fazem parte do corpus, porém foram desprezadas para fins de treinamento dos classificadores, juntamente com instâncias que apresentavam erros de árvore sintática detectados posteriormente por meios automáticos..

O que se observou nessa primeira versão do Propbank-Br, que está disponível para download no Portal PortLex, é que muitos verbos tinham apenas uma instância de anotação e essa esparsidade de dados era muito indesejável para fins de aprendizado de máquina.

Visando criar um corpus de treinamento maior, iniciamos um novo projeto de anotação de papéis semânticos. O novo corpus é chamado de Propbank-Br.v2, sendo também do gênero jornalístico, extraído do corpus PLN-Br FULL (Bruckschen et al., 2008).

Este relatório descreve a seleção de sentenças para compor o corpus Propbank-Br-v2, que está disponível para download no site do Projeto Prosa.

Como não existe outro *treebank* do português além do Bosque, já anotado no primeiro projeto Propbank-Br, a nova anotação foi realizada sobre árvores sintáticas não corrigidas. Essa característica levou-nos a conceber alguns filtros para rejeitar as sentenças que não possuem árvores sintáticas bem formadas.

Quando se trata de aprender anotação de papéis semânticos, temos que considerar três variáveis:

1. quais verbos estão representados no corpus
2. quais sentidos dos verbos estão representados no corpus
3. quais alternâncias sintáticas estão representadas no corpus

A noção de que um verbo pode ter vários sentidos é largamente difundida, porém o que se percebeu na prática é que não existem limites claros entre um sentido e outro. A divisão de sentidos pode ter diferentes graus de detalhamento e deve ser feita levando em conta sua finalidade. Para fins de anotação de papéis semânticos, o detalhamento ideal é aquele que está associado a marcas na superfície do texto, como o uso de diferentes preposições, por exemplo.

Cada sentido do verbo pode admitir diferentes alternâncias sintáticas. Alternância é mudança da ordem dos constituintes (sintáticos, semânticos ou ambos ao mesmo tempo). Por exemplo, a voz passiva (exemplos 1 e 2) é uma alternância sintática que marca uma alternância semântica (o paciente ou tema toma o lugar do agente na posição de sujeito sintático). Mas um sujeito oculto (exemplos 3 e 4) ou um sujeito posposto (exemplos 5 e 6), por outro lado, são alternâncias na ordem dos constituintes que não correspondem a alternâncias de papéis sintáticos (o sujeito sintático continua o mesmo).

1. João feriu José. (Agente V Paciente)
2. José foi ferido por João. (Paciente V Agente)
3. Eu consegui entender a matéria. (Agente V Tema)
4. Consegui entender a matéria. (V Tema)
5. Empenham-se, os alunos, para conseguir bons resultados. (V Agente Finalidade)
6. Os alunos empenham-se para conseguir bons resultados (Agente V Finalidade)

Via de regra, a quantidade de sentidos de um verbo está associada à quantidade de alternâncias sintáticas que ele admite, mas podem existir verbos que admitem grande número de alternâncias para um mesmo sentido.

Nosso objetivo, portanto, é evitar a esparsidade de dados e fornecer um corpus de treinamento com diversidade de verbos, de sentidos de verbos e de alternâncias sintáticas admitidas pelos sentidos dos verbos. Isso seria possível de ser obtido com um grande corpus anotado. Nosso problema, porém, é conseguir o máximo de representatividade nos três itens com o menor número de instâncias de anotação, já que a anotação manual é dispendiosa.

Nossa estratégia foi separar as alternâncias altamente frequentes e reduzir sua participação no corpus de anotação (para não gastarmos recursos humanos com tarefas repetitivas). Essa separação, por outro lado, permitiu que evidenciássemos as alternâncias menos frequentes ou menos previsíveis, ou seja, aquelas que desafiam mais os classificadores.

Exemplo de verbo: RECLAMAR

Exemplos de sentidos de verbo:

RECLAMAR.01:

alguém (agente) queixa-se sobre algo (tópico) para alguém (ouvinte)

João reclamou do preço da mensalidade para o diretor da escola.

RECLAMAR.02:

alguém (agente) reivindica um direito (tema)

Maria reclamou a pensão dos filhos na justiça.

Exemplos de alternâncias de um sentido do verbo (RECLAMAR.01):

Eles reclamaram ao professor quanto ao prazo para apresentação dos trabalhos

Todos reclamam sobre a falta de ética do governo.

Ele reclamou dela para mim.

De falta de trabalho ninguém reclama.

Alternâncias sintáticas podem ser expressas genericamente pelo que conhecemos como *subcategorization frames* (SCFs), como vemos a seguir para os exemplos de alternâncias mostradas acima:

NP reclamar [a] NP [quanto a] NP

NP reclamar [sobre] NP

NP reclamar [a respeito de] NP [para] NP

[de] NP NP reclamar

A experiência acumulada ao longo da primeira versão do Propbank-Br foi utilizada para definir critérios (padrões) a seleção das sentenças que deveriam compor o novo corpus de anotação que atendessem a requisitos desejáveis para um corpus de treinamento. O custo da mão de obra de anotação linguística é alto e, por isso, é importante que os anotadores humanos sejam aproveitados para anotar fenômenos variados e não repetitivos. Embora certa quantidade de fenômenos repetidos seja desejável, pois o material anotado será usado para aprendizado de máquina, é interessante ter algum controle dessa repetição.

Na anotação de papéis semânticos, selecionam-se instâncias para anotação a partir de palavras que possuem estruturas argumentais, ou seja, palavras que "evocam" outras palavras, também chamadas de palavras eventivas. As classes de palavras eventivas são: verbo (todos, com exceção dos auxiliares), nomes (alguns, como por exemplo "autorização", que possui um agente, um tema), adjetivos (alguns) e advérbios (alguns). No nosso projeto, estamos trabalhando apenas com verbos, que são palavras eventivas por excelência. Os verbos, contudo, podem ter muitos sentidos e cada um deles "evoca" uma estrutura argumental diferente. É de nosso interesse anotar uma mesma quantidade de instâncias de cada sentido de cada verbo. Por exemplo, o verbo "esperar" tem dois grandes sentidos: o de "aguardar" e o de "ter esperança". Como garantir que as instâncias de anotação tenham uma participação proporcional de cada um?

Segundo vasto conhecimento acumulado pela semântica, apoiado na hipótese de Levin (Levin, 1993), percebeu-se que a mudança de sentido dos verbos está relacionada a mudanças na estrutura sintática das orações, ou seja, existe correlação entre sintaxe e semântica. Nossa

experiência, em português, nos levou a observar quais as principais marcas sintáticas que indicam mudança de sentido de verbos e usamos essas marcas para selecionar as instâncias de anotação para a criação da versão 2 do corpus Propbank-Br.

Temos dois objetivos para a criação do corpus: limitar a quantidade de exemplos dos sentidos identificáveis e altamente frequentes e aumentar a amostra de instâncias de anotação para capturar sentidos pouco frequentes e que não são identificáveis sintaticamente.

Este relatório está organizado em três seções. A Seção 2 apresenta o processo de criação de grupos de sentenças candidatas com fenômenos linguísticos distintos, a Seção 3 apresenta o processo de seleção de sentenças representativas para anotação e a Seção 4 a geração de instâncias de anotação.

2. CRIAÇÃO DE GRUPOS DE SENTENÇAS CANDIDATAS COM FENÔMENOS LINGUÍSTICOS DISTINTOS

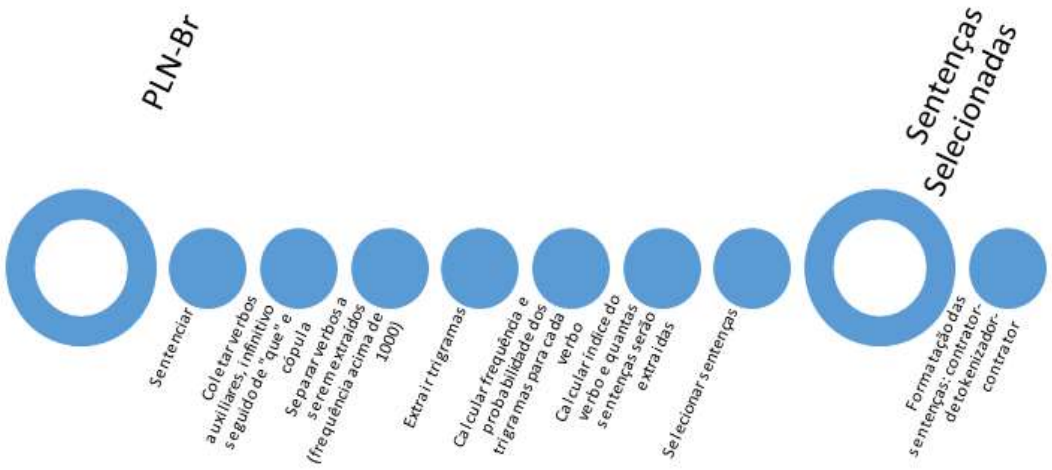

Figura 1: Seleção de sentenças candidatas para anotação

Uma série de sete passos foi concebida para fazer a seleção de sentenças candidatas para anotação (Figura 1). Essa série foi implementada computacionalmente por Lianet Sepúlveda Torres e o resultado de cada uma delas foi analisado por Magali Sanches Duran, a fim de corrigir possíveis desvios.

O corpus PLN-BR FULL **(**Bruckschen et. al., 2008**)**, que contém 103.080 mil textos da Folha de São Paulo e 29.014.089 tokens, foi anotado automaticamente com o parser Palavras (Bick, 2000) e disponibilizado no formato MOSES. Esse corpus contém 1.567.227 sentenças e 4.159.046 ocorrências de verbos que, lematizadas, representam 9.927 verbos.

Cada sentença do corpus pode ser selecionada *n* vezes para o corpus de anotação, uma vez para cada um dos *n* verbos que nela ocorrem. A sentença selecionada para anotação de um determinado verbo é chamada de instância. Portanto, como o PLN-BR tem 4.159.046 ocorrências de verbos, ele pode gerar até 4.159.046 instâncias de anotação.

Esse número, contudo, exigiria um esforço de anotação incomensurável e, por isso, é preciso selecionar uma amostra representativa desse universo que contenha, na medida do possível, os mesmos sentidos de verbos contidos no corpus total. Após analisar a lista de lemas de verbos do corpus, concluímos que os Hapax legomena (verbos com uma única ocorrência no corpus), que somam 3.814 verbos (38% do corpus), constituem erros, salvo raríssimas exceções. Os verbos com duas, três e quatro ocorrências no corpus também apresentam muitos erros, pois todas as palavras não reconhecidas pelo dicionário do parser são classificadas como verbos.

A fim de eliminar candidatos espúrios e garantir que o corpus de treinamento não tivesse esparsidade de dados, todos os verbos com frequência < 5 foram desprezados. Sobraram, após esse corte, 4497 verbos. Se um verbo tinha 5 instâncias para anotação, todas foram selecionadas. Se tinha de 6 a 100 ocorrências, as instâncias entraram no processo de seleção de instâncias por modelo de língua (SÉTIMO PASSO da estratégia de seleção).

Faixa de Frequência	Quantidade de Lemas
1 a 4	5439
5 a 10	861
11 a 50	1363
51 a 100	568
101 a 500	920
501 a 1000	244
Acima de 1000	541

Tabela 1. Quantidade de verbos por faixa de frequência.

Na Tabela 1 apresentamos a quantidade de verbos por faixa de frequência no corpus PLN-Br. A faixa de frequência em amarelo foi excluída do corpus de anotação e a faixa de frequência em verde foi priorizada para anotação. Todas as demais faixas contêm sentenças relevantes para anotação e poderão ser objeto de anotações futuras.

PRIMEIRO PASSO

Transformar as sentenças em instâncias. Para cada verbo da sentença deverá ser criada uma instância de anotação. Por exemplo, a sentença:

> Ele queria fazer uma experiência, mas o chefe não permitiu.

gera três instâncias de anotação:

QUERER 2 Ele **queria** fazer uma experiência, mas o chefe não permitiu.

FAZER 3 Ele queria **fazer** uma experiência, mas o chefe não permitiu.

PERMITIR 10 Ele queria fazer uma experiência, mas o chefe não **permitiu.**

É importante guardar:

- Identificador da sentença no corpus original
- Identificador do verbo alvo da anotação (posição do verbo alvo na sentença), pois um mesmo exemplo pode conter duas ocorrências do mesmo verbo e cada um será alvo de anotação em uma instância diferente.

SEGUNDO PASSO

O segundo passo foi fazer uma redução do número de instâncias elegíveis para anotação, eliminando todas as instâncias de verbos auxiliares. Verbos auxiliares são aqueles que não possuem estrutura argumental e que modificam outros verbos, auxiliando-os a expressar tempo, modo, aspecto e diátese (de voz passiva). Temos uma tabela para identificar esses verbos (Duran e Aluisio, 2010), pois eles não são auxiliares sempre, mas apenas quando usados em determinadas condições. Incluímos na tabela padrões com o advérbio de negação "não" antes do auxiliado, após perceber os padrões da tabela podem apresentar material interveniente (o mais comum é o "não") entre o verbo auxiliar e o verbo auxiliado.

Por exemplo, o verbo DEVER, seguido de outro verbo no infinitivo, é auxiliar (1), mas em outras condições não o é (2).

(1) Ele deve fazer plantão esta noite.

(2) Ele deve mais de dez mil reais no banco.

Os verbos auxiliares, embora muito importantes para a semântica, podem ser identificados com o uso de regras e, portanto, não necessitam ser submetidos ao aprendizado de máquina e, pelo mesmo motivo, não necessitam de anotação manual. Eles estão entre os 100 verbos mais frequentes do corpus. Fazendo essa exclusão, eliminamos 529.205 instâncias de nosso corpus de possíveis instâncias de anotação.

Requirement	Input corpus (A)	Instances meeting the requirement (B)	Instances Selected for annotation (C)	Remaining instances eligible under this requirement (B-C)	Output corpus (A-B)
V + que	3.619.064	148.247	1.226	147.021	3.470.817
V + INF	3.470.817	98.246	823	97.423	3.372.571
Copula verbs	3.372.571	551.891	150	551.653	2.820.680

Refl + V or V + Refl	2.820.680	71.333	10.819	60.514	2.749.505
Total		869.717	13.018	856.699	

TERCEIRO PASSO: VERBOS SEGUIDOS DE "QUE"

O terceiro passo foi selecionar um número fixo de sentenças de todos os verbos seguidos da conjunção integrante QUE, ou seja, que possuem um argumento sob forma de oração subordinada desenvolvida. Verbos que atendem a essa condição são altamente freqüentes no corpus, por se tratar de corpus jornalístico e grande parte deles serem verbos de elocução (dizer que, falar que, afirmar que, contar que, etc.). De acordo com nossa experiência prévia, um verbo seguido de infinitivo tem um único sentido. De 148.247 instâncias que atendiam a esta condição, selecionamos 1.226, reservamos 147.021 para seleções futuras e o corpus ficou reduzido a 3.470.817 instâncias. A fim de evitarmos seleções equivocadas, selecionamos apenas as instâncias em que o verbo não está no particípio (PCP), pois o Palavras classifica como Verbos os Adjetivos formados por particípio. Ex: Compre um papel quadriculado que é melhor.

QUARTO PASSO: VERBOS SEGUIDOS DE VERBO NO INFINITIVO

O quarto passo é selecionar um número fixo de sentenças de verbos seguidos de outros verbos na forma infinitiva. De acordo com nossa experiência prévia, um verbo seguido de infinitivo tem um único sentido. É importante que esse passo ocorra após remoção de todos os auxiliares, pois muitos auxiliares são seguidos de infinitivo. Incluir as seguintes restrições:

- Desconsiderar o verbo SER. O verbo SER é desconsiderado por se tratar de cópula e ser usado com inversão de foco: O que ele quer é fazer isso.
- Desconsiderar exemplos em que o V está na forma PCP (particípio), visto que o particípio, nesses casos, tem quase sempre um papel de adjetivo e não de verbo. Os particípios são desconsiderados, pois o Palavras não distingue particípios nas funções de substantivo e de adjetivo, classificando-os todos como verbo. Exemplo: É importante para o aposentado fazer economia.
- considerar o padrão V + INF
- considerar o padrão V "não" INF

Das 98.246 instâncias que atendem essa condição, selecionamos 823 para anotação, reservamos 97.423 e o corpus ficou reduzido a 3.372.571 instâncias.

QUINTO PASSO

O quinto passo foi selecionar 30 instâncias de verbos de cópula (ser, estar, ficar, permanecer, parecer). Os demais verbos de cópula não foram contemplados neste passo, pois são polissêmicos e nenhuma etiqueta sintática foi identificada para fazer a seleção automática. Os verbos de cópula, embora sejam pouco polissêmicos, admitem uma grande variedade de formas de argumentos. Por exemplo:

Ele permanece cansado.

Ele permanece em estado de choque.

Ele permanece com a mesma cara de sempre.

Ele permanece de mau humor.

Ele permanece aqui apesar de não gostar.

Das 551.891 instâncias dos cinco verbos de cópula pesquisados, selecionamos 150 instâncias para anotação, reservamos 551.741 e o corpus ficou reduzido a 2. 820.680 instâncias. Uma vez eliminados os verbos auxiliares e de cópula, podemos enxergar os verbos verdadeiramente polissêmicos do corpus.

SEXTO PASSO: VERBOS PRONOMINAIS

O objetivo deste passo foi selecionar as ocorrências de verbos acompanhados de pronomes reflexivos e que estão na tabela de verbos pronominais. Como o uso dos pronomes reflexivos junto a esses verbos pode ter diferentes funções, o trabalho de desambiguização constitui grande desafio para a anotação e exige conhecimento linguístico (Duran et al. 2013).

Possuímos uma tabela dos verbos que mudam de sentido quando pronominalizados (Duran, 2013). Essa tabela foi utilizada para selecionar as instâncias de verbos pronominalizados.

Para os verbos da tabela foram selecionadas 20 instâncias de para anotação, sendo 10 instâncias de verbos seguidos por pronome reflexivo (-me, -te, -se, -nos, -vos) e 10 instâncias de verbos antecedidos por pronome reflexivo (me, te, se, nos, vos). Se havia menos de 10 instâncias que

atendessem as condições, todas foram selecionadas. Observação: todos os reflexivos (< refl>) são PERS (pronomes pessoais), mas nem todo PERS é <refl>. Essa informação sintática auxiliou-nos a prefiltrar os candidatos.

Das 71.333 instâncias que atendiam a essas condições, selecionamos 10.819 instâncias para anotação, reservamos 60.514 e o corpus ficou reduzido a 2.749.505 instâncias, as quais correspondem aos verbos cuja polissemia desconhecemos. Assumindo a quantidade de trigramas em um modelo de língua esteja relacionada à quantidade de alternâncias sintáticas e que a quantidade de alternâncias sintáticas esteja relacionada ao número de sentidos dos verbos, utilizamos o modelo de língua para selecionar nesse corpus de 2.749.505 instâncias uma amostra significante para anotação.

SÉTIMO PASSO: MODELO DE LÍNGUA

Os scripts estão alocados na pasta selectSentenceByVerbsInfo/ no servidor de experimentos do NILC. O código está separado em 3 pastas.

1. **implementacao** → Nesta pasta estão os scripts criados para executar o procedimento. Os scripts .sh que estão dentro desta pasta são utilizados para executar cada passo do procedimento de forma independente. Além disso, o script rodaExperimento.sh executa todo o procedimento. Os scripts sh executam os scripts em python que estão na pasta "selectSenteceByVerbsInfo"
2. **saida** → Contém todas as saídas do procedimento. A saída de cada passo do procedimento é armazenada nas pastas que estão dentro da pasta saída. Por exemplo, a saída do passo para extrair os verbos auxiliares é armazenada na pasta "passo_2_Auxiliares". Antes de executar o experimento, devem ser criadas as pastas de cada passo e dentro delas não deve aparecer nenhum arquivo.
3. **resources** → pasta com arquivos criados pela Magali e outros recursos lexicais necessários no processo de seleção
    a. auxiliarVerbs_New → verbos auxiliares e padrões
    b. erroVerbs → anotação manual da Magali que indica erros na anotação de verbos realizada pelo Palavras
    c. verbosPronominais → lista de verbos pronominais
4. **palavrasPLN_Br** → A pasta está no servidor de experimentos em "palavrasPLN_Br/". Nela foram armazenados o corpus processado pelo palavras e além disso, o corpus original.

Lista de scripts

plnBr_VerbsS.py → Gera lista de verbos ordenada pela frequência. A saída deste script é colocada na pasta principal.

Entrada: (i) corpus anotado pelo palavras com a saída do moses; (ii) lista com erros de verbos

Saída: lista dos verbos que aparecem no corpus e sua frequência

searchElement_Bigcorpus.py → Após obter a lista de verbos e sua frequência os verbos do corpus são indexados. A indexação do corpus, consiste em obter para cada um dos verbos uma lista com todas as sentenças em que ele acontece. A tarefa principal do procedimento descrito neste relatório é a anotação semântica de diferentes sentidos dos verbos, sendo necessário analisar todas as instâncias dos verbos. Desta forma, as sentenças em que mais de um verbo aconteceu foram duplicadas.

Entrada: (i) Corpus anotado pelo palavras; (ii) verbos listados por frequências

Saída: arquivo com a informação estruturada como a seguir:

Verbo+Frequência+Lista das sentenças em que aconteceu o verbo_posição do verbo na sentença

A seguir um exemplo. O verbo "prover" aparece no corpus 6 vezes e acontece na sentença que está na posição 51277 e o verbo está na posição 8 dentro da sentença. A posição da sentença dentro do corpus é o identificador utilizada para identificar as sentenças.

Verbo	**Frequência**	**Número da sentença_Posição do verbo na sentença**

prover	6	51277_8	333204_1	333205_14	390457_6

Passo 2: Eliminar verbos auxiliares

Neste passo são eliminadas do corpus todas as sentenças nas quais o verbo indexado é auxiliar. Para identificar um verbo é auxiliar foi adotado um conjunto de padrões, criados manualmente pela Magali. Esses padrões estão listados no arquivo "auxiliarVerbs_New" que está dentro da pasta "resources".

Entrada: (i) lista com os padrões dos auxiliares; **(ii)** lista dos verbos indexados; **(iii)** corpus anotado pelo palavras;

Saída: **(i)** arquivo com a lista de sentenças em que o verbo indexado era um auxiliar; **(ii)** arquivo com as sentenças em que o verbo indexado não tinha padrão de auxiliar (novo corpus); (iii) estatísticas do passo de eliminação de auxiliares.

selectSentenceByVerbsInfo/saida/passo_2_Auxiliares

Entrada	index_VerbsPosSentence → arquivo com o corpus original indexado plnbr-parsed.txt.moses → corpus anotado pelo Palavras auxiliarVerbs_New → arquivo com a lista de verbos auxiliares
Saída	freqVerb_AUX → arquivo com as estatísticas da seleção dos auxiliares index_NewCorpus → indexação das instâncias dos verbos que não seguiam os padrões dos auxiliares index_SentenceAUX → indexação com as instâncias que seguem os padrões dos auxiliares

Construção do corpus de anotação

Nos passos apresentados a seguir são selecionadas as sentenças para criar o corpus de anotação. Nos passos 3,4,5 e 6 foram selecionadas aquelas instâncias em que os verbos seguiam um padrão

específico. Todas as instâncias que cumprem com o padrão requisitado são eliminadas do corpus original e são armazenadas em arquivos independentes como explicado a seguir.

Em cada passo é criado um arquivo, no qual aparecem "x" sentenças para anotar.

Se o número de instâncias do verbo que cumprem o padrão for menor que o número de instâncias requisitadas, então todas as instâncias são selecionadas.

Se o número de instâncias for maior que o número requisitado, as instâncias excedentes são armazenadas separadamente. → Arquivo com as sentenças armazenadas

Se a sentença em que acontece o padrão tiver mais de 25 palavras essas sentenças são eliminadas da análise e armazenadas em um arquivo separado. → Sentenças eliminadas

Em geral, as saídas dos passos 3,4,5 e 6 são listadas a seguir:

Entradas

Passo 3 → verbos seguidos de "QUE" (KS → Anotação do Palavras) Verbs_QUE.py	index_NewCorpus → o novo corpus indexado após eliminar os auxiliares (passo 2) plnbr-parsed.txt.moses → corpus anotado pelo palavras
Passo 4 → V+INF Verbs_Inf.py	index_NewCorpus → o novo corpus indexado após eliminar as instâncias dos verbos+QUE (passo 3) plnbr-parsed.txt.moses → corpus anotado pelo palavras
Passo 5 → Verbos Copula	index_NewCorpus → o novo corpus indexado após eliminar os verbos+INF (passo 4)

Verbs_Copula.py	plnbr-parsed.txt.moses → corpus anotado pelo palavras
Verbos pronominais pronominalVerbs.py	index_NewCorpus → o novo corpus indexado após eliminar as instâncias dos verbos de cópula plnbr-parsed.txt.moses → corpus anotado pelo palavras
Saída	**index_NewCorpus** → indexação do novo corpus, no qual foram eliminadas as instâncias que cumpriam alguns dos padrões anteriores **freqVerb_ → ...** → estatísticas com a quantidade de sentenças que cumpriram o padrão e foram eliminadas do corpus original e a quantidade de sentenças que não cumpriam o padrão e continuam no corpus. index_Verbs … → Arquivo com a indexação das sentenças que cumprem com os padrões anteriores
buildCorpus.py	
Saída	**estatisticaSelecao_....** → arquivo com as estatísticas das instâncias selecionadas, armazenadas e eliminadas em cada passo; **index_anotar** … → Indexação com as sentenças selecionadas para anotar **index_armazenadas** … → Indexação com as sentenças que foram armazenadas **index_eliminadas** … → Indexação com as sentenças que foram eliminadas

Problemas na seleção das sentenças de anotação. Por exemplo:

O verbo “procurar” acontece em 7 sentenças com o padrão V+QUE(KS). Como o verbo tem mais de 5 exemplos, então essas sentenças são selecionadas para anotação. Acontece que 5

dessas sentenças têm mais de 25 palavras, desta forma somente um exemplo do verbo procurar foi selecionado. A mesma coisa acontece com o verbo “obviar”.

Comando para gerar a contagem de frequência do modelo de língua

./ngram-count -order 3 -text /home/lianet/LM/corpusPasso_6Cru -no-sos -no-eos -write-vocab /home/lianet/LM/plnBR_Vocab -lm /home/lianet/LM/plnBR_LM_New -unk -write3 /home/lianet/LM/plnBR_3Vocab

3. SELEÇÃO DE SENTENÇAS REPRESENTATIVAS PARA ANOTAÇÃO

Esta etapa tem como objetivo separar uma amostra representativa de sentenças do corpus. Durante o processo descrito pela Figura 1, as sentenças foram separadas em classes de acordo com os seus critérios linguísticos. Nesta etapa, trabalhamos com as sentenças pertencentes a um agrupamento no qual se faz necessária a aplicação de técnicas de aprendizagem de máquina na tarefa de anotação de papéis semântico. Dessa categoria, escolhemos iniciar o processo de anotação pelos verbos mais frequentes e com maior chance de serem encontrados como entrada pelo classificador. O processo de seleção da amostra representativa está ilustrado pela Figura 2 e os resultados desta etapa serão utilizados para a geração das instâncias de anotação, ilustrado na Figura 3. A Figura 4 traz o processo completo de seleção e preparação de sentenças do corpus PLN-Br para compor o corpus de anotação de papéis semânticos Propbank-Br.v2.

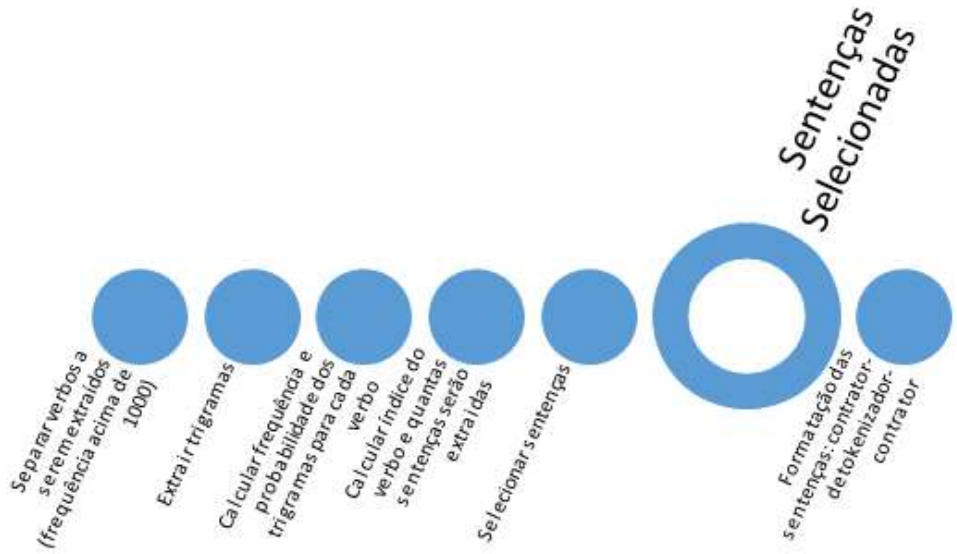

Figura 2: Seleção de sentenças representativas para anotação

3.1 SEPARANDO VERBOS COM FREQUÊNCIA ACIMA DE 1000

Das instâncias que restaram após os PASSOS de 1 a 5 acima descritos, fizemos uma estatística para descobrir qual a faixa de frequência de verbos continha 90% das instâncias. Essa estatística serviu para decidirmos selecionar os verbos com frequência acima de 1000 (541 verbos). Se esses verbos representam 90% das ocorrências do corpus, assumimos que eles constituem um bom material para treinamento de um classificador de papéis semânticos, pois se o classificador aprender a classificar bem 90% do corpus ele deverá ter uma boa precisão. Além disso, os verbos

altamente frequentes, excluindo-se auxiliares, verbos de cópula e verbos de complemento oracional (V+que e V+ INF), são muito provavelmente os mais polissêmicos da língua. A estratégia de priorização de verbos com frequência acima de 1000 teve por objetivo gerar um corpus de treinamento o mais rápido possível e com baixa esparsidade de dados.

Outro motivo que nos levou a fazer esse recorte inicial é o fato de que os anotadores têm que consultar um repositório verbal para fazer a anotação e queríamos aproveitar o feedback dos anotadores para revisar o repositório. Se trabalhássemos com muitos verbos ao mesmo tempo, o trabalho de revisão do repositório ficaria comprometido, já que estava a cargo de uma única linguista. Os demais verbos do corpus, ou seja, aqueles com frequência abaixo de 1.000, não foram desprezados, pois pretendíamos anotá-los em uma etapa futura do projeto.

A listagem dos verbos com frequência acima de 1000 é feita pelo script **separar_over_1000.py**. Esse script trabalha com a lista de verbos localizada em '*passo_6_V_Pronominais/freqVerb_AUX+QUE+INF+Copula+Pronominal*' e cria um arquivo que contém uma lista de verbos com frequência acima de 1000 seguidos de suas frequências. O Apêndice A traz a lista com o resultado deste script. O arquivo que contém essa lista está localizado na pasta *'/verbFreqover1000/over1000.txt'*

Há um script similar, chamado '/separar_arq.py" que separa os verbos entre uma dada faixa de frequência e pode ser utilizado para coletar as de menor frequência.

3.2 Extração de trigramas e cálculo de sua probabilidade e frequência

Para cada verbo da lista com frequência acima de 1000, precisamos coletar os trigramas com o verbo na primeira posição, fornecendo o contexto à direta, para trazer a variação de contextos de uso destes (mais detalhes são dados na Seção 3.3). O script para esta coleta é "trigramas_freq.py", localizado em "scriptsMarina/" e é responsável pela coleta dos trigramas e por calcular sua frequência e probabilidade. Ele utiliza os scripts "readImportantNGram.py" e "jointPb_NGram_Freq.py", os quais selecionam trigramas e calculam sua frequência dado um verbo. O script "trigramas_freq.py" recebe uma lista de verbos e suas frequências (como o arquivo "over1000.txt") e executa "readImportantNGram.py" e "jointPb_NGram_Freq.py" para todos os verbos da lista.

trigramas_freq.py
Entrada =>
1. Arquivo contendo lista de verbos => Ex: over1000.txt
2. Programa "readImportantNGram.py"
3. Programa "jointPb_NGram_Freq.py"
4. ../modeloLianet/saida/LM/plnBR_LM
5. ../modeloLianet/saida/LM/plnBR_3Vocab
Saida =>
1. Indicar uma pasta para output. Nessa pasta será colocado um arquivo por verbo da lista, cada um contendo os trigramas do respectivo verbo seguidos de sua probabilidade e frequência.

Observações Importantes:

1. O script demora um tempo considerável para coletar os trigramas dos verbos. Para 515 verbos o script leva aproximadamente 1 dia e meio para coletar os trigramas no corpus PLN-BR. Sua complexidade depende do numero de verbos a serem processados e do tamanho do corpus.

2. Os scripts readImportantNGram.py e jointPb_NGram_Freq.py coletam trigramas a partir do modelo de língua. Porém, o modelo de língua contém trigramas que já foram selecionados em etapas anteriores. Essa duplicação será compensada posteriormente durante o script de seleção de sentenças, descrito na Seção 3.4.

3.3 Calcular índice do verbo e número de sentenças a serem coletadas

Para cada verbo precisamos decidir o número de sentenças a serem coletadas. Nossa decisão foi baseada em um índice calculado da seguinte maneira:

índice = número de trigramas diferentes/frequência do verbo

Dessa maneira, conseguimos identificar uma razão que expressa uma riqueza de contextos para cada verbo. O número de sentença coletadas para cada verbo é:

número de sentenças coletadas = índice * 100

O script que calcula o índice é chamado "calcula_indice.py" e ele é responsável por ordenar os trigramas por frequência, calcular seus índices e determinar o número de sentenças a serem coletadas para cada verbo.

calcula_indice.py
Entrada =>
1. Arquivo contendo lista de verbos => Ex: over1000.txt
2. Pasta contendo os arquivos dos verbos contendo os trigramas. Ex: *'/home/marina/scriptsMarina/verbosEscolhidos1000/'*
Saida =>
1. Arquivos de trigramas ordenado
2. Arquivo da lista de verbos alterado, com a adição do índice e **número** de trigramas a serem coletados por cada verbo

Observação: O número mínimo de sentenças a serem coletadas para cada verbo é 10, mesmo se o índice indicar um número inferior.

3.4 Seleção de Sentenças

O objetivo da etapa de seleção de sentenças é, dado um número de sentenças a serem coletadas para um verbo como parâmetro, coletar uma sentença de cada um dos trigramas mais frequentes. O tamanho da sentença também foi utilizado como parâmetro de corte. As sentenças com tamanho superior a 30 *tokens* também não foram coletadas, pois além de serem de difícil compreensão para humanos, são mais propensas a gerar uma análise sintática com erros pelo parser PALAVRAS, o que foi atestado com vários testes com sentenças de tamanhos variados.

calcula_indice.py
Entrada =>
1. Arquivo contendo lista de verbos, tendo em cada linha: frequencia, número de trigramas, índice, número de sentenças a serem coletadas => Ex: over1000.txt
2. Pasta contendo os arquivos dos verbos contendo os trigramas. Ex: '*scriptsMarina/verbosEscolhidos1000/*'
3. Lista de índices para as sentenças no corpus. Ex: *'modeloLianet/saida/passo_6_V_Pronominais/index_NewCorpus'*
4. Corpus sentenciado e tokenizado pelo PALAVRAS. Ex: *'/plnbr_Sentences'*
5. Corpus com caixa baixa e verbo lematizado. Ex: "*palavrasPLN_Br/plnBR_cru*'
6. Pasta onde serão colocados os arquivos de cada verbo com suas sentenças
Saida =>
Na pasta especificada no item 6 serão criados um arquivo por verbo. Em cada um dos arquivos estarão as sentenças.
O formato das sentenças é: YYYXXXXXX [Sentença], sendo que YYY é o indice do verbo e XXXXXX (de tamanho variável) é o índice da sentença. [Sentença] é o conteúdo da sentença.

Observações: Esse script também demora um tempo significativo em sua execução. Ele lida com o fato de que os trigramas encontrados na etapa descrita na Seção 1.2 podem ter sido separados previamente (ex: verbos de cópula) e não constem mais no índice. Portanto, quando há um pedido de 20 trigramas para o índice, ele pode retornar menos. Quando ele retorna menos, o script faz mais um pedido para conseguir os restantes e assim por diante. Só virá menos sentenças do que o determinado quando o arquivo de trigramas tiver siso inteiramente percorrido, por exemplo, com somente 14 das 20 sentenças encontradas.

3.5 Detokenização e Contração

Durante a etapa de sentenciação do corpus, as sentenças originais não foram indexadas. Portanto, ao final da etapa de seleção das sentenças não conseguimos obter a sentença original. No entanto, como o parser PALAVRAS trabalha com sentenças naturais (originais) como entrada, foi necessária a construção de um detokenizador que reverte o processo de tokenização.

Nesta etapa, primeiro executamos um script chamado contrator, desenvolvido por Erick Maziero, sobre as sentenças tokenizadas. No entanto, o contrator não prevê sentenças tokenizadas, portanto é necessário desativar o módulo de tokenização/detokenização. Ao desativar esse módulo e executar o contrator antes do detokenizador, o contrator é capaz de contrair muitos casos com facilidade. Após a sentenças serem contraídas é preciso detokenizar. Porém, com a quebra de multiwords feita pelo detokenizador, aparecem mais palavras a serem contraídas. Portanto, passamos o contrator novamente no corpus.

O detokenizador tem as seguintes funcionalidades:

- Quebrar multiwords
- Trocar aspas duplas por aspas simples
- Tratar cifrão
- Tratar problemas de aspas não pareadas:
  - Se há uma aspa não pareada seguida de vírgula, inserimos aspas no início da sentença
  - Aspas não pareadas são excluídas
- Tratar asteriscos não pareados.
- Tratar parenteses, chaves e colchetes não pareados
- Junta pontuações com a palavra à esquerda (por exemplo: “,”, “.” “!”, “?”)
- Junta demais tipos de pontuação com palavras à direita (por exemplo, #, $)
- Tratamento de travessões: palavra “TRAVESSÃO” é substituída por um "-"
- Contração de crase
- Contrações de -la, -lo, -las, -los, -lhe(s), -a, -o, -se
- Retirar caracteres que provocam quebra de sentença (por exemplo, “:”, “;”)

detok_nathan.py
Entrada => python detok_nathan.py -i /pasta_input/ -o /pasta_output/
1. Comando: -i Pasta contendo arquivos de sentenças tokenizadas.
2. Comanda: -o Pasta para receber as sentenças detokenizadas.
Saida =>
1. Arquivos contendo sentenças detokenizadas

4. GERAÇÃO DE INSTÂNCIAS DE ANOTAÇÃO

Nesta seção será apresentado o procedimento para execução do parser sintático PALAVRAS sobre as sentenças selecionadas do corpus PLN-Br a fim de gerar instâncias para anotação humana de papéis semânticos.

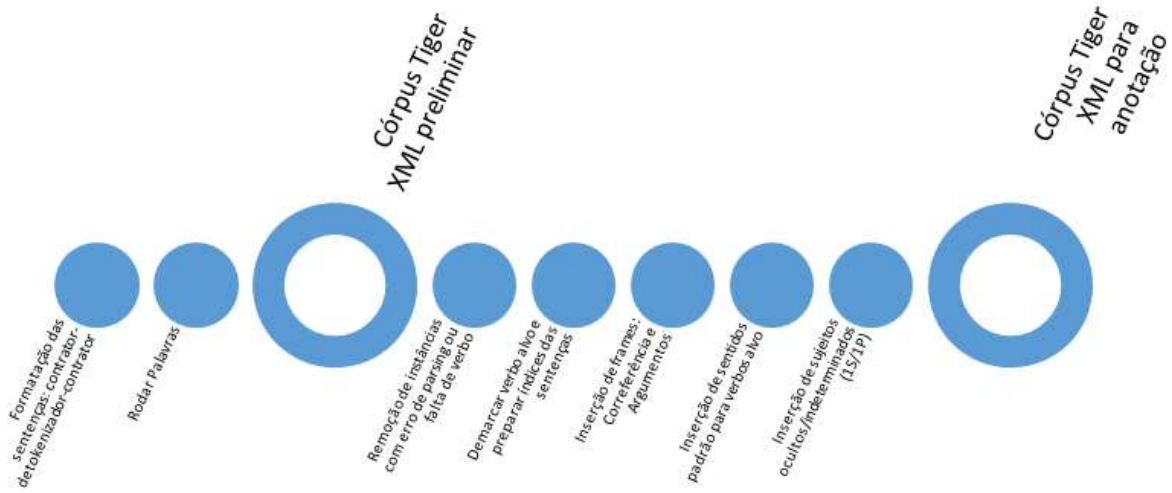

Figura 3: Geração de instâncias de anotação

4.1 PRÉ-REQUISITOS

O parser sintático Palavras, em sua execução, não trabalha corretamente com sentenças que possuam elementos sem contração (por exemplo, na = em + a). As sentenças selecionadas do corpus PLN-Br, no entanto, não possuem contrações devido a uma restrição do sentenciador previamente utilizado na geração do corpus. Elas também são sentenças tokenizadas, devido ao cálculo de trigramas previamente realizado à esta etapa. Logo, é necessário realizar um procedimento para retornar as sentenças às suas formas originais (sem marcas de tokenização e possuindo contrações). Após contrair as sentenças selecionadas do corpus e remover marcas de tokenização, as sentenças obtidas se encontram no seguinte formato, composto por metadados no formato numérico, seguidos da sentença em questão:

0062852 Cerca de 500 pessoas foram obrigadas a abandonar suas casas.

O numeral que marca o início de cada sentença é composto pela posição do verbo alvo na sentença e pelo ID original da sentença no corpus PLN-Br, possibilitando a recuperação da instância original do corpus. A posição do verbo é composta pelos 3 primeiros digitos e o ID do verbo pelos restantes. No exemplo apresentado acima, o verbo é o sétimo token da sentença (contagem a partir de zero), e a sentença é a 2853ª do corpus PLN-Br. Vale ressaltar que, para a

contagem da posição do verbo, a sentença deve estar tokenizada, ou seja, símbolos de pontuação contam como um token.

Como os metadados do início de cada sentença, se passados pelo parser, estariam contidos na árvore sintática da sentença (o que é indesejável para o anotador humano), é necessário isolar essa informação. Para tanto, foi desenvolvido o script preprocessing_id_verb_sentences.py, na linguagem de programação Python, que processa os arquivos de sentenças contidos em um diretório alvo, extraindo os metadados de cada sentença. A seguir uma breve descrição das operações de entrada e saída do script desenvolvido.

Entrada: O script espera três parâmetros de entrada: *dir_input*, *dir_ids* e *dir_sentences*

dir_input é o diretório em que se encontram os arquivos que contém as sentenças juntamente com seus metadados. Corresponde aos arquivos que deverão ser processados

dir_ids é o diretório que o usuário deseja que os metadados sejam salvos. O numeral de uma sentença será gravado em um arquivo com o mesmo nome do arquivo original, facilitando sua recuperação.

dir_sentences é o diretório que o usuário deseja que as sentenças sejam salvas. As sentenças serão gravadas em um arquivo com o mesmo nome do arquivo original, facilitando sua recuperação.

Saída: todos os metadados serão salvos no diretório *dir_ids,* em arquivos com o mesmo nome dos contidos no diretório dir_input

1. todas as sentenças, sem seus metadados, serão salvos no diretório *dir_sentences*, em arquivos com o mesmo nome dos contidos no diretório *dir_input*.

Separando as sentenças de seus metadados (que possibilitam identificar qual o verbo a ser anotado na sentença e qual a posição da sentença no corpus original), é possível executar o parser Palavras.

4.2 A EXECUÇÃO

Para gerar as instâncias de anotação, devemos passar cada sentença pelo parser PALAVRAS. Foi desenvolvido o script run_palavras_plnbr.py, na linguagem de programação Python, que executa o parser PALAVRAS sobre um diretório de arquivos texto, retornando um mesmo número de arquivos que o fornecido como entrada. Definimos que cada arquivo deve corresponder às instâncias de um verbo, mas poderia haver apenas um arquivo com todas as instâncias de todos os verbos desejados. Ao executar esse script sobre as sentenças selecionadas do corpus PLN-Br, temos as sentenças no formato XML. A seguir, uma breve descrição das operações de entrada e saída do script:

Entrada: O script espera um parâmetro de entrada: *dir_sentences*

dir_sentences é o diretório que contém arquivos com as sentenças, sem metadados, que devem ser passadas pelo parser Palavras. Corresponde ao diretório *dir_sentences* gerado como output do *script preprocessing_id_verb_sentences.py*.

Saída: são geradas três saídas desse script.

parcial é um diretório que contém as árvores sintáticas, em formato XML, geradas pelo parser Palavras.

output é um diretório que contém as árvores sintáticas sem erros e preparadas para anotação (a ser explicado adiante), preparadas para anotação.

log.txt é um arquivo que marca todas as ocorrências de erros encontrados nas árvores sintáticas. Esses erros podem ser devido à falta de verbo em uma sentença ou por uma árvore que contém ao menos um elemento órfão dos demais, ou seja, que não possui ligação.

4.3 PÓS-PROCESSAMENTO DO XML

Além de gerar as instâncias XML da árvore sintática das sentenças fornecidas, o script run_palavras_plnbr.py realiza uma série de atividades para detectar erros nas árvores sintáticas e para inserir tags XML necessárias para a anotação do corpus na ferramenta Salto.

1. Após geradas as instâncias, são eliminadas todas aquelas que não contenham verbo de anotação, que o verbo de anotação seja um verbo auxiliar ou que possuam uma árvore mal formada. Consideramos uma árvore mal formada, os casos em que existe ao menos um elemento não ligado ao restante da árvore. Esse caso é conhecido como grafo desconexo ou grafo não conexo.
2. O segundo passo executado é demarcar o verbo alvo da anotação no XML, com a utilização do metadado que marca a posição do verbo na sentença (presente no diretório dir_ids gerado como saída do script preprocessing_id_verb_sentences.py). A tag XML inserida é a <fenode>, dentro da tag <target>, contendo o valor idref referente ao token do verbo. No exemplo a seguir, o idref=”2852_7” refere-se ao verbo na posição 7, da sentença 2852.

```
<sem>
    <frames>
        <frame name="Argumentos">
            <target>
                <fenode idref="2852_7"/>
            </target>
```

```
            </frame>

        </frames>
        <wordtags>
            <wordtag idref="2852_7" name="sentido">abandonar.00</wordtag>
        </wordtags>
    </sem>
```

3. O terceiro passo executado é renomear os IDs do XML de cada sentença. A renomeação dos IDs do XML de uma sentença é feita com a utilização do metadado da sentença em questão que demarca o ID da sentença no corpus PLN-Br. Assim, todo campo ID do XML da sentença estará relacionado ao índice da sentença no corpus original. A seguir, é apresentado os elementos XML inseridos para demarcar o verbo alvo a ser anotado.

4. O quarto passo executado é a inserção dos frames *Argumentos* e *Correferência* no cabeçalho do arquivo XML (<*head*>). O primeiro frame demarca todos os argumentos previstos para anotação de papéis semânticos (por exemplo, Arg0, Arg1, ArgM-Tmp). O segundo demarca os elementos que correferenciam sujeitos previamente citados. A identificação de correferentes é feita segundo a *tag part of speech* "pron-indp", fornecida pelo parser Palavras. Essa inserção é feita para que o anotador humano possa relacionar os correferentes com seus referentes, criando um corpus de correferência para trabalhos futuros. O *frame* completo de Argumentos e Correferente é apresentado a seguir.

```
<frames xmlns="http://www.clt-st.de/framenet/frame-database">
    <frame name="Argumentos">
        <element name="argm-loc" optional="false"/>
        <element name="argm-tmp" optional="false"/>
        <element name="argm-ext" optional="true"/>
        <element name="argm-mnr" optional="false"/>
        <element name="argm-prp" optional="true"/>
        <element name="argm-cau" optional="false"/>
```

```
            <element name="argm-dir" optional="true"/>
            <element name="argm-dis" optional="true"/>
            <element name="arg0" optional="false"/>
            <element name="arg1" optional="false"/>
            <element name="arg2" optional="false"/>
            <element name="argm-neg" optional="false"/>
            <element name="arg3" optional="true"/>
            <element name="argm-adv" optional="true"/>
            <element name="argm-prd" optional="true"/>
            <element name="arg4" optional="true"/>
            <element name="argm-rec" optional="true"/>
            <element name="arg5" optional="true"/>
            <element name="argm-exp" optional="true"/>
            <element name="argm-asp" optional="true"/>
            <element name="argm-mod" optional="true"/>
            <element name="argm-pas" optional="true"/>
            <element name="argm-tml" optional="true"/>
            <element name="argm-com" optional="true"/>
            <element name="argm-gol" optional="true"/>
            <element name="argm-nse" optional="true"/>
        </frame>
        <frame name="Correferente">
                <element name="referente" optional="false"/>
         </frame>
</frames>
```

5. O quinto passo realizado pelo *script* é a inserção automática de sentidos padrão para os verbos alvo. Foi definido que o sentido padrão de um verbo é o lema desse verbo, seguido de ".00". Por exemplo, o sentido padrão do verbo "abandonar" é "abandonar.00", como apresentado a seguir. Essa ação deve minimizar o esforço humano na anotação do sentido do verbo, uma vez que este deverá apenas alterar o número do sentido, mantendo o lema do verbo escrito.

```
<wordtags>

    <wordtag idref="2852_7" name="sentido">abandonar.00</wordtag>

</wordtags>
```

6. O sexto e último passo realizado é a inserção automática de sujeitos ocultos. Por exemplo:

   Gostei muito disso → Eu gostei muito disso

   Após estudos, foi decidido inserir apenas sujeitos ocultos nas primeiras pessoas do singular e plural (eu e nós), devido à acurácia obtida em estudos, que foi baixa para as terceiras pessoas. A inserção dos sujeitos ocultos aumentará a ocorrência de anotação, principalmente, de Arg0 (agente da ação do verbo). Como os sistemas automáticos de anotação de papéis semânticos seguem uma ordem de anotação, dando prioridade para o agente da ação, uma vez que esse elemento for erroneamente anotado, aumenta-se a chance de anotação errada dos demais argumentos. Assim, percebe-se a necessidade de explicitar esses sujeitos: aumentar a ocorrência de Arg0 no corpus anotado.

Uma vez que foram realizadas essas seis etapas, os arquivos XML estão prontos para anotação humana.

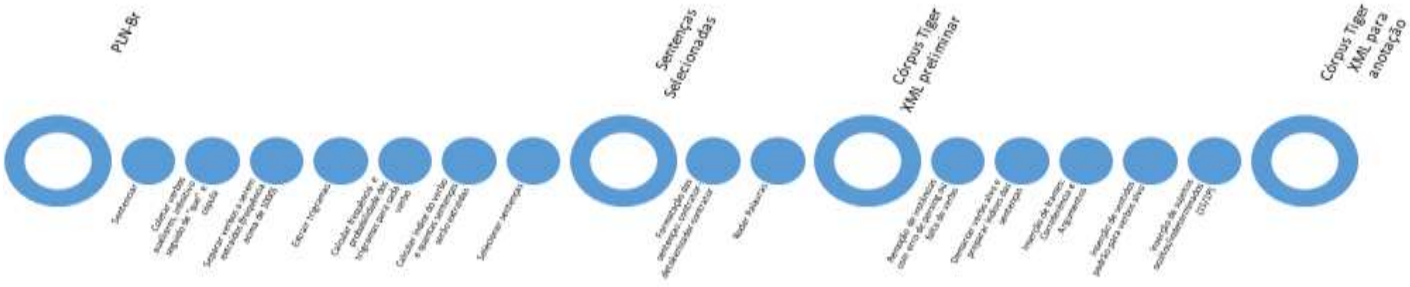

FIGURA 4: PROCESSO COMPLETO DE SELEÇÃO E PREPARAÇÃO DE SENTENÇAS DO CORPUS PLN-BR PARA COMPOR O CORPUS DE ANOTAÇÃO DE PAPÉIS SEMÂNTICOS PROPBANK-BR.V2

REFERÊNCIAS

Afonso S. ; Bick, E. ; Haber, E. ; Santos, D. (2002) Floresta sintá(c)tica: a treebank for Portuguese. In: Proceedings of LREC-2002. Disponível em: http://www.linguateca.pt/documentos/AfonsoetalLREC2002.pdf

Bick, E. (2000). The Parsing System Palavras Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus, Denmark, Aarhus University Press.

Bruckschen, M., Muniz, F., Souza, J. G. C., Fuchs, J. T., Infante, K., Muniz, M., Gonçalves, P. N., Vieira, R. e Aluísio, S. M. (2008): Anotação Lingüística em XML do Corpus PLN-BR. Série de Relatórios do NILC. NILC-TR-09-08, 39 p.

Duran, M. S.; Aluísio, S. M. (2011) Propbank-Br: a Brazilian Portuguese corpus annotated with semantic role labels. In the Proceedings of the 8th Symposium in Information and Human Language Technology, October 24-26, Cuiabá/MT, Brazil, pg. 164–168.

Duran, M. S. ; ALUÍSIO, S. M. . Propbank-Br: a Brazilian Treebank annotated with semantic role labels. In: Eight International Conference on Language Resources and Evaluation (LREC'12), Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis (editors), 2012, Istambul. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). Paris: European Language Resources Association (ELRA), 2012. v. 1. p. 1862-1867.

Duran, M. S.; Aluísio, S. M. (2010). Verbos Auxiliares no Português do Brasil. Série de Relatórios Técnicos do Núcleo Interinstitucional de Linguística Computacional, NILC-TR-10-05, 25 pgs.

Duran, M.S.; Scarton, C. E.; Aluísio, S.M. ; Ramisch, C. (2013). Identifying pronominal verbs: Towards automatic disambiguation of the clitic 'se' in Portuguese. In Proceedings of the 9th Workshop on Multiword Expressions, pages 93–100, Atlanta, Georgia, USA, June.

Levin, B. (1993): English Verb Classes and Alternation, A Preliminary Investigation. The University of Chicago Press.

APÊNDICE A

Tabela de verbos com mais de 1000 de frequência. Os dados estão dispostos da seguinte maneira: verbo, frequência do verbo, número de trigramas, índice, número de sentenças

armar 2291 310 0.14 14.0	calcular 1359 254 0.19 19.0
acordar 1025 185 0.18 18.0	atacar 3137 743 0.24 24.0
correr 4281 581 0.14 14.0	presidir 1231 200 0.16 16.0
tratar 9659 1045 0.11 11.0	ter 125952 24060 0.19 19.0
separar 1600 300 0.19 19.0	participar 9413 636 0.07 7.0
desenvolver 3631 917 0.25 25.0	enfrentar 6693 1682 0.25 25.0
ver 23303 4727 0.2 20.0	aproximar 1674 251 0.15 15.0
reforçar 1989 441 0.22 22.0	valer 3320 488 0.15 15.0
operar 2489 537 0.22 22.0	emitir 1380 275 0.2 20.0
viajar 3154 517 0.16 16.0	recuperar 1916 666 0.35 35.0
retirar 3315 757 0.23 23.0	exportar 1163 255 0.22 22.0
acumular 2941 403 0.14 14.0	colaborar 1708 204 0.12 12.0
fazer 88935 12506 0.14 14.0	restar 1258 161 0.13 13.0
exigir 4350 836 0.19 19.0	variar 2284 276 0.12 12.0
empatar 1294 146 0.11 11.0	pôr 2909 395 0.14 14.0
existir 10804 1483 0.14 14.0	custar 4443 558 0.13 13.0
vestir 1453 306 0.21 21.0	destinar 2831 333 0.12 12.0
afirmar 22262 1943 0.09 9.0	preparar 4759 853 0.18 18.0
alcançar 2413 616 0.26 26.0	atrair 2142 681 0.32 32.0

possuir 3674 564 0.15 15.0	comprovar 1313 344 0.26 26.0
atingir 8371 1528 0.18 18.0	baixar 1516 347 0.23 23.0
casar 1920 289 0.15 15.0	desistir 1449 188 0.13 13.0
funcionar 5089 739 0.15 15.0	acertar 2640 514 0.19 19.0
decidir 4283 1032 0.24 24.0	evitar 5743 2501 0.44 44.0
reconhecer 2915 676 0.23 23.0	somar 2394 347 0.14 14.0
exibir 3468 551 0.16 16.0	causar 5658 1033 0.18 18.0
incorporar 1067 192 0.18 18.0	localizar 2806 423 0.15 15.0
mencionar 1138 251 0.22 22.0	substituir 3100 779 0.25 25.0
dirigir 3692 679 0.18 18.0	vir 12940 1412 0.11 11.0
querer 8608 1676 0.19 19.0	ampliar 2311 594 0.26 26.0
entender 3848 1151 0.3 30.0	terminar 4895 798 0.16 16.0
estimar 1993 236 0.12 12.0	excluir 1336 231 0.17 17.0
provar 1084 423 0.39 39.0	unir 1358 312 0.23 23.0
gravar 2848 540 0.19 19.0	executar 1256 304 0.24 24.0
provocar 6026 1092 0.18 18.0	comparecer 1077 105 0.1 10.0
trazer 8487 1464 0.17 17.0	comemorar 1954 434 0.22 22.0
aplicar 3786 754 0.2 20.0	adquirir 2343 607 0.26 26.0
prever 9215 1076 0.12 12.0	comentar 3184 618 0.19 19.0
decretar 1078 158 0.15 15.0	obrigar 3773 627 0.17 17.0
realizar 10926 1818 0.17 17.0	apostar 1512 171 0.11 11.0
descrever 1681 340 0.2 20.0	construir 4190 1147 0.27 27.0
continuar 7153 1885 0.26 26.0	perder 12518 2134 0.17 17.0
mexer 1036 172 0.17 17.0	conduzir 1688 308 0.18 18.0

incluir 8263 1272 0.15 15.0	relatar 1211 206 0.17 17.0
perceber 1761 502 0.29 29.0	interromper 1404 327 0.23 23.0
prejudicar 2559 480 0.19 19.0	concordar 2297 247 0.11 11.0
garantir 5535 1304 0.24 24.0	surpreender 1295 225 0.17 17.0
pertencer 2177 133 0.06 6.0	conferir 1388 352 0.25 25.0
destruir 2376 551 0.23 23.0	abrigar 1694 482 0.28 28.0
bater 3873 657 0.17 17.0	estudar 3889 907 0.23 23.0
melhorar 3945 766 0.19 19.0	colocar 10643 1901 0.18 18.0
recolher 1361 354 0.26 26.0	importar 3226 435 0.13 13.0
levar 18566 2874 0.15 15.0	retomar 2517 437 0.17 17.0
supor 3160 484 0.15 15.0	inaugurar 1833 312 0.17 17.0
fabricar 1076 231 0.21 21.0	ligar 7093 635 0.09 9.0
trocar 3311 905 0.27 27.0	aproveitar 2859 751 0.26 26.0
facilitar 1797 402 0.22 22.0	implantar 1423 325 0.23 23.0
dedicar 2286 246 0.11 11.0	descartar 1754 229 0.13 13.0
tirar 4639 1164 0.25 25.0	extinguir 1020 198 0.19 19.0
iniciar 4510 832 0.18 18.0	atrapalhar 1030 249 0.24 24.0
circular 1298 275 0.21 21.0	elevar 2356 442 0.19 19.0
distribuir 2797 580 0.21 21.0	voltar 9833 1073 0.11 11.0
aguardar 1670 321 0.19 19.0	corrigir 1319 366 0.28 28.0
editar 1101 200 0.18 18.0	segurar 1071 302 0.28 28.0
preferir 1927 324 0.17 17.0	proibir 2747 427 0.16 16.0
sentir 4640 997 0.21 21.0	demorar 1612 327 0.2 20.0
começar 11364 1520 0.13 13.0	solicitar 1640 349 0.21 21.0

controlar 3519 835 0.24 24.0	buscar 3657 1110 0.3 30.0
mudar 7323 1328 0.18 18.0	desejar 1409 270 0.19 19.0
sofrer 6954 1129 0.16 16.0	passar 20881 2485 0.12 12.0
convencer 1871 593 0.32 32.0	ameaçar 2281 413 0.18 18.0
aliar 1068 106 0.1 10.0	treinar 2040 468 0.23 23.0
ferir 1785 313 0.18 18.0	afastar 2610 405 0.16 16.0
enviar 4759 726 0.15 15.0	surgir 4153 577 0.14 14.0
dar 34840 4977 0.14 14.0	emprestar 1182 201 0.17 17.0
gritar 1171 249 0.21 21.0	testar 1184 324 0.27 27.0
selecionar 1146 241 0.21 21.0	vencer 7189 1332 0.19 19.0
completar 3312 622 0.19 19.0	dever 5949 867 0.15 15.0
subir 6199 1040 0.17 17.0	temer 1434 469 0.33 33.0
perguntar 3655 566 0.15 15.0	acionar 1006 251 0.25 25.0
determinar 4605 819 0.18 18.0	promover 4530 1040 0.23 23.0
abordar 1280 220 0.17 17.0	visitar 2912 683 0.23 23.0
cercar 1028 155 0.15 15.0	citar 4033 650 0.16 16.0
nomear 1051 223 0.21 21.0	escapar 1538 234 0.15 15.0
matar 6533 1308 0.2 20.0	reservar 1408 186 0.13 13.0
precisar 4726 792 0.17 17.0	procurar 6103 1363 0.22 22.0
sugerir 2077 384 0.18 18.0	envolver 6701 828 0.12 12.0
praticar 2474 492 0.2 20.0	organizar 3251 677 0.21 21.0
encaminhar 2177 288 0.13 13.0	agir 2330 555 0.24 24.0
constar 1304 128 0.1 10.0	financiar 1971 594 0.3 30.0
convocar 2385 481 0.2 20.0	roubar 1737 341 0.2 20.0

devolver 1176 256 0.22 22.0	dispor 2895 412 0.14 14.0
analisar 3601 808 0.22 22.0	conquistar 3044 690 0.23 23.0
acontecer 13714 1420 0.1 10.0	orientar 1270 339 0.27 27.0
influenciar 1038 193 0.19 19.0	insistir 1317 181 0.14 14.0
entrar 14056 1000 0.07 7.0	aprovar 7109 935 0.13 13.0
jogar 9152 1707 0.19 19.0	forçar 1416 396 0.28 28.0
cair 8820 1351 0.15 15.0	alterar 2072 521 0.25 25.0
acrescentar 1811 302 0.17 17.0	divulgar 7568 1034 0.14 14.0
instalar 3081 600 0.19 19.0	dançar 1137 253 0.22 22.0
detectar 1100 320 0.29 29.0	atravessar 1087 247 0.23 23.0
admitir 1970 451 0.23 23.0	registrar 6876 1006 0.15 15.0
interpretar 1760 318 0.18 18.0	deixar 14561 2915 0.2 20.0
isolar 1569 241 0.15 15.0	prender 4996 730 0.15 15.0
carregar 1645 330 0.2 20.0	adotar 4184 828 0.2 20.0
retornar 1647 167 0.1 10.0	suspender 3338 594 0.18 18.0
avaliar 3493 743 0.21 21.0	render 1680 312 0.19 19.0
repassar 2894 362 0.13 13.0	misturar 1640 275 0.17 17.0
haver 52190 7343 0.14 14.0	encerrar 2387 419 0.18 18.0
gerar 4117 1001 0.24 24.0	fixar 3411 479 0.14 14.0
contribuir 2403 294 0.12 12.0	disparar 1347 215 0.16 16.0
aceitar 4892 1160 0.24 24.0	reclamar 2361 285 0.12 12.0
concorrer 1538 227 0.15 15.0	observar 2002 466 0.23 23.0
eleger 4709 616 0.13 13.0	assistir 3026 423 0.14 14.0
abrir 12539 1664 0.13 13.0	tender 1706 193 0.11 11.0

respeitar 1606 351 0.22 22.0	apoiar 3702 785 0.21 21.0
conhecer 10453 1710 0.16 16.0	fundar 1077 203 0.19 19.0
pegar 3977 1000 0.25 25.0	salvar 1581 509 0.32 32.0
inspirar 1320 135 0.1 10.0	nascer 3609 490 0.14 14.0
aumentar 8884 1512 0.17 17.0	autorizar 2293 391 0.17 17.0
associar 1185 132 0.11 11.0	cobrar 4679 826 0.18 18.0
crescer 6395 1103 0.17 17.0	revelar 4005 733 0.18 18.0
recuar 1379 262 0.19 19.0	ignorar 1046 245 0.23 23.0
responder 5022 744 0.15 15.0	quebrar 2470 469 0.19 19.0
romper 1224 253 0.21 21.0	bastar 1324 148 0.11 11.0
tornar 3757 1432 0.38 38.0	receber 21676 3937 0.18 18.0
arrecadar 1559 322 0.21 21.0	convidar 2093 333 0.16 16.0
prestar 3112 475 0.15 15.0	modificar 1147 279 0.24 24.0
superar 3024 744 0.25 25.0	conceder 3167 512 0.16 16.0
elaborar 1887 349 0.18 18.0	tentar 3244 2709 0.84 84.0
caber 2157 172 0.08 8.0	lançar 8202 1185 0.14 14.0
defender 7633 1468 0.19 19.0	disputar 5808 952 0.16 16.0
atuar 4960 761 0.15 15.0	explicar 6040 1223 0.2 20.0
sustentar 1621 418 0.26 26.0	corresponder 1377 123 0.09 9.0
ouvir 6947 1556 0.22 22.0	sentar 1136 140 0.12 12.0
anunciar 6900 984 0.14 14.0	acusar 6313 745 0.12 12.0
depender 4468 256 0.06 6.0	liderar 2605 343 0.13 13.0
pensar 5679 1086 0.19 19.0	explorar 1493 485 0.32 32.0
sobreviver 1268 255 0.2 20.0	confirmar 4696 771 0.16 16.0

relacionar 1998 224 0.11 11.0	vender 8798 1761 0.2 20.0
coordenar 1207 213 0.18 18.0	achar 4445 913 0.21 21.0
soltar 1105 202 0.18 18.0	indicar 4867 825 0.17 17.0
significar 2545 423 0.17 17.0	pagar 15684 2741 0.17 17.0
explodir 1000 206 0.21 21.0	adiar 1677 367 0.22 22.0
assumir 5838 954 0.16 16.0	trabalhar 12048 1891 0.16 16.0
descer 1050 209 0.2 20.0	renunciar 1100 183 0.17 17.0
comprar 8259 2168 0.26 26.0	classificar 1976 303 0.15 15.0
partir 3046 282 0.09 9.0	combater 1933 590 0.31 31.0
dispensar 1141 192 0.17 17.0	refletir 1797 345 0.19 19.0
olhar 2245 782 0.35 35.0	errar 2312 253 0.11 11.0
acreditar 3051 473 0.16 16.0	eliminar 2234 628 0.28 28.0
especializar 1220 169 0.14 14.0	cortar 2346 623 0.27 27.0
cancelar 1138 264 0.23 23.0	justificar 2497 657 0.26 26.0
planejar 1188 288 0.24 24.0	considerar 9239 1675 0.18 18.0
escrever 7579 1141 0.15 15.0	expor 2141 402 0.19 19.0
demonstrar 1975 472 0.24 24.0	ler 7281 1129 0.16 16.0
contar 9954 1359 0.14 14.0	desconhecer 1324 228 0.17 17.0
dividir 3948 626 0.16 16.0	utilizar 5385 1040 0.19 19.0
merecer 1601 240 0.15 15.0	concluir 2703 529 0.2 20.0
identificar 3250 909 0.28 28.0	estrear 2527 292 0.12 12.0
esclarecer 1275 392 0.31 31.0	comer 2295 735 0.32 32.0
marcar 8218 1120 0.14 14.0	imaginar 1882 497 0.26 26.0
descobrir 2979 859 0.29 29.0	juntar 2768 404 0.15 15.0

aposentar 1021 210 0.21 21.0	andar 2214 655 0.3 30.0
dominar 1910 345 0.18 18.0	acabar 5683 860 0.15 15.0
sair 13014 1405 0.11 11.0	avançar 1398 262 0.19 19.0
conseguir 7056 2115 0.3 30.0	basear 2168 172 0.08 8.0
transferir 2324 399 0.17 17.0	dizer 64136 6477 0.1 10.0
concentrar 1558 309 0.2 20.0	apurar 3104 581 0.19 19.0
medir 1620 305 0.19 19.0	entregar 4671 799 0.17 17.0
competir 1001 213 0.21 21.0	reunir 7156 1123 0.16 16.0
afetar 2521 417 0.17 17.0	recomendar 1502 225 0.15 15.0
complicar 1182 153 0.13 13.0	compor 2846 562 0.2 20.0
aparecer 5733 773 0.13 13.0	atribuir 1994 290 0.15 15.0
produzir 6417 1437 0.22 22.0	faltar 3637 497 0.14 14.0
constituir 1497 304 0.2 20.0	derrubar 1916 464 0.24 24.0
recorrer 2240 267 0.12 12.0	cuidar 1932 184 0.1 10.0
conversar 2876 467 0.16 16.0	administrar 1685 430 0.26 26.0
submeter 1999 241 0.12 12.0	ensinar 1393 366 0.26 26.0
propor 4140 715 0.17 17.0	assegurar 1366 419 0.31 31.0
desaparecer 1413 204 0.14 14.0	denunciar 1653 354 0.21 21.0
verificar 1778 467 0.26 26.0	favorecer 1238 282 0.23 23.0
lutar 1940 293 0.15 15.0	deter 1863 366 0.2 20.0
governar 1024 267 0.26 26.0	guardar 1446 349 0.24 24.0
pressionar 1943 396 0.2 20.0	repetir 2933 601 0.2 20.0
destacar 1222 255 0.21 21.0	cantar 2167 466 0.22 22.0
combinar 1071 186 0.17 17.0	processar 1151 325 0.28 28.0

apresentar 14790 2369 0.16 16.0	reduzir 6376 1239 0.19 19.0
mandar 2729 681 0.25 25.0	interessar 2035 252 0.12 12.0
ocorrer 12638 1440 0.11 11.0	durar 2188 445 0.2 20.0
comprometer 1151 309 0.27 27.0	preocupar 2186 280 0.13 13.0
parar 3925 861 0.22 22.0	impedir 3375 949 0.28 28.0
resultar 1933 228 0.12 12.0	encontrar 10432 2253 0.22 22.0
ajudar 6881 1319 0.19 19.0	chamar 7406 1021 0.14 14.0
informar 5883 838 0.14 14.0	servir 5637 927 0.16 16.0
fechar 8750 1133 0.13 13.0	atirar 1290 223 0.17 17.0
resistir 1285 150 0.12 12.0	declarar 4377 682 0.16 16.0
morar 3505 408 0.12 12.0	ir 20541 2067 0.1 10.0
tomar 10235 1675 0.16 16.0	falar 14418 2228 0.15 15.0
formar 4945 940 0.19 19.0	assassinar 1292 201 0.16 16.0
esconder 1773 387 0.22 22.0	sobrar 1035 145 0.14 14.0
aprender 2576 604 0.23 23.0	alimentar 1182 448 0.38 38.0
resolver 3690 698 0.19 19.0	cobrir 2183 508 0.23 23.0
viver 8424 1287 0.15 15.0	conter 2793 675 0.24 24.0
adequar 1426 218 0.15 15.0	mover 1337 194 0.15 15.0
movimentar 1294 249 0.19 19.0	depor 1276 276 0.22 22.0
negar 3694 609 0.16 16.0	pedir 12462 1927 0.15 15.0
diminuir 3201 754 0.24 24.0	oferecer 7814 1469 0.19 19.0
abandonar 2527 562 0.22 22.0	publicar 7218 740 0.1 10.0
pesquisar 1014 207 0.2 20.0	contratar 3526 731 0.21 21.0
estimular 1804 474 0.26 26.0	cotar 1187 85 0.07 7.0

compensar 1234 356 0.29 29.0	optar 1870 259 0.14 14.0
cometer 2772 481 0.17 17.0	inventar 1078 242 0.22 22.0
consultar 1233 319 0.26 26.0	representar 6396 1020 0.16 16.0
gostar 6538 580 0.09 9.0	pretender 1037 214 0.21 21.0
rever 1122 410 0.37 37.0	restringir 1073 198 0.18 18.0
pesar 1347 185 0.14 14.0	acelerar 1133 288 0.25 25.0
permitir 5358 946 0.18 18.0	prometer 1901 254 0.13 13.0
morrer 7596 958 0.13 13.0	apreender 1187 180 0.15 15.0
beneficiar 2022 411 0.2 20.0	expulsar 1372 200 0.15 15.0
rejeitar 1488 249 0.17 17.0	levantar 2187 472 0.22 22.0
compreender 1168 421 0.36 36.0	esquecer 1784 444 0.25 25.0
apontar 4773 705 0.15 15.0	manter 13735 3060 0.22 22.0
mostrar 9656 2063 0.21 21.0	percorrer 1012 243 0.24 24.0
definir 6320 1344 0.21 21.0	gastar 3382 698 0.21 21.0
criticar 3464 616 0.18 18.0	integrar 2502 481 0.19 19.0
cruzar 1128 212 0.19 19.0	projetar 1559 248 0.16 16.0
comparar 2329 384 0.16 16.0	chegar 20504 1637 0.08 8.0
preservar 1289 464 0.36 36.0	obter 7055 1654 0.23 23.0
montar 2945 716 0.24 24.0	transformar 2859 885 0.31 31.0
virar 4112 897 0.22 22.0	questionar 2467 430 0.17 17.0
comandar 2570 423 0.16 16.0	limitar 2279 352 0.15 15.0
consumir 1518 311 0.2 20.0	estabelecer 3383 814 0.24 24.0
fornecer 2288 489 0.21 21.0	escolher 4123 1050 0.25 25.0
transmitir 1964 429 0.22 22.0	lidar 1109 202 0.18 18.0

investir 4127 748 0.18 18.0	firmar 1045 192 0.18 18.0
atender 4962 875 0.18 18.0	discutir 5226 1303 0.25 25.0
usar 16791 3395 0.2 20.0	reagir 1585 266 0.17 17.0
tocar 4018 785 0.2 20.0	exercer 1658 343 0.21 21.0
impor 2604 497 0.19 19.0	poder 8794 4269 0.49 49.0
avisar 1239 225 0.18 18.0	votar 4660 710 0.15 15.0
ocupar 4679 739 0.16 16.0	dormir 1507 350 0.23 23.0
acompanhar 6372 1167 0.18 18.0	brincar 1385 281 0.2 20.0
esperar 6816 1289 0.19 19.0	ultrapassar 1468 303 0.21 21.0
criar 12582 2806 0.22 22.0	programar 1320 206 0.16 16.0
seguir 7854 1335 0.17 17.0	punir 1374 364 0.26 26.0
julgar 2477 561 0.23 23.0	condenar 2925 458 0.16 16.0
ceder 1564 296 0.19 19.0	desviar 1137 184 0.16 16.0
alegar 1028 183 0.18 18.0	invadir 2048 376 0.18 18.0
notar 1056 229 0.22 22.0	proteger 2319 702 0.3 30.0
antecipar 1196 317 0.27 27.0	liberar 2965 613 0.21 21.0
cumprir 4467 792 0.18 18.0	saber 16190 2630 0.16 16.0
elogiar 1112 203 0.18 18.0	lembrar 4212 919 0.22 22.0
assinar 5141 695 0.14 14.0	entrevistar 1079 165 0.15 15.0
negociar 4975 919 0.18 18.0	derrotar 2195 393 0.18 18.0
ganhar 12527 2188 0.17 17.0	caminhar 1115 225 0.2 20.0
fugir 2772 331 0.12 12.0	demitir 1587 303 0.19 19.0
investigar 4272 838 0.2 20.0	