Automatische toekenning werkwoordstijden

Bjork M. Westmeijer (b.m.westmeijer@students.uu.nl)
studentennummer: 5502187

25-08-2018
Bachelor Kunstmatige Intelligentie, UU
7.5 ECTS

Begeleiders:
Martijn van der Klis & Henriëtte de Swart

Tweede beoordelaar:
Marnix Naber

Inhoudsopgave

Samenvatting

In dit onderzoek is onderzocht of het mogelijk is om automatisch werkwoordstijden toe te kennen aan werkwoordscombinaties uit datasets. Er zijn vijf talen onderzocht: Nederlands, Spaans, Engels, Frans en Duits. De nadruk lag echter op Nederlands en Spaans. Er is onderzocht of een beslisboom kon worden getraind op een dataset en vervolgens worden gevalideerd op een andere dataset. Een handmatig opgesteld regelgebaseerd algoritme is gebruikt als basislijn voor het Nederlands en het Spaans. Uit de resultaten is gebleken dat voor het Nederlands een percentage kon worden behaald van boven de 90 procent, voor het Spaans rond de 70 procent. Het algoritme behaalde slechtere resultaten dan het regelgebaseerde algoritme voor beide talen.

Inleiding

Het doel van dit werkstuk is om het toekennen van werkwoordstijden te automatiseren. Neem de volgende zin: *Jantje **is** naar school **gefietst***. Aan de combinatie van de gemarkeerde woorden wordt een bepaalde tijd gegeven, in deze zin de voltooid tegenwoordige tijd. Doordat het handmatig annoteren van werkwoordstijden veel tijd in beslag neemt en daarbij ook wel eens fouten worden gemaakt, wordt er hier een poging gedaan om dit door middel van een algoritme te automatiseren.

Dit eindwerkstuk is een onderdeel van het grotere Time in Translation project. Met dit eindwerkstuk wordt geprobeerd een deel van het dataverwerkingsproces te automatiseren, waardoor de data sneller en hopelijk ook met minder handmatige fouten verwerkt kan worden. Op dit moment worden werkwoordstijden handmatig geannoteerd in de datasets. Deze datasets zijn deelverzamelingen van een corpus. Een corpus is een tekst of tekstenverzameling die gebruikt kan worden voor onderzoek. In een dataset is daarnaast ook aanvullende annotatie die nodig kan zijn voor verder onderzoek.

De twee talen waar dit onderzoek zich voornamelijk op richt zijn het Nederlands en het Spaans omdat deze twee talen verschillende structuren hebben voor het spreken over de verleden tijd. In figuur 1 en 2 staan er Nederlandse en Spaanse voorbeeldzinnen met de bijbehorende werkwoordstijden.

Figuur 1: Voorbeeldzinnen Nederlands

1. *Jan heeft vannacht op de bank geslapen.* (v.t.t.)
2. *Piet liep gisteren naar huis.* (o.v.t.)
3. *Jan was eerder gekomen dan Petra.*(v.v.t.)
4. *Jan loopt naar huis.*(o.t.t.)

Tussen het Nederlands en het Spaans zijn er een aantal verschillen:

- Het Nederlands heeft alleen de onvoltooide verleden tijd, terwijl het Spaans de pretérito indefinido en imperfecto heeft. Beide vormen hebben hun eigen gebruik. De pretérito indefinido wordt veel gebruikt om te spreken over

Figuur 2: Voorbeeldzinnen Spaans en bijbehorende vertalingen
1. García come una manzana.(presente)
2. Ayer Rodrigo fue a la escuela en bici.(pretérito indefinido)
3. Cada sábado ibamos a la casa de nuestro abuelo.(pretérito imperfecto)
4. Esta mañana he recibido una carta.(presente perfecto compuesto)
5. Quizás el lector sepa hablar español.(presente de subjuntivo)
6. Vi una estrella la última noche.(pretérito indefinido)
7. No sabía que eran las doce.(pretérito imperfecto)
1. García eet een appel.(o.t.t.)
2. Gisteren ging Rodrigo op de fiets naar school.(o.v.t.)
3. Elke zaterdag gingen we naar het huis van onze opa.(o.v.t.)
4. Vanochtend heb ik een brief gekregen.(v.t.t.)
5. Misschien kan de lezer Spaans spreken.(o.t.t.)
6. Ik zag afgelopen nacht een ster.(o.v.t.)
7. Ik wist niet dat het twaalf uur was.(o.v.t.)

eenmalige gebeurtenissen in het verleden. De imperfecto wordt voornamelijk gebruikt om te praten over beschrijvingen van situaties en omgevingen, gewoontes, tijd, kennis en contrasten met het heden.

- In het Spaans wordt het subjunctief gebruikt, in het Nederlands komt de aanvoegende wijs nauwelijks voor. Het subjunctief wordt voornamelijk gebruikt om te spreken over subjectieve zaken of wanneer er onzekerheid is.

Daarentegen is het het doel dat het algoritme ook werkt als er werkwoordstijden moeten worden toegekend aan andere talen. Het Duits, Engels en Frans worden om deze reden ook in het onderzoek betrokken. In verband met tijdslimitaties zullen de analyses van deze talen minder diepgaand zijn.

De onderzoeksvraag die is opgesteld voor dit werkstuk is: Kunnen er automatisch werkwoordstijden worden toegekend aan zinnen?
De twee deelvragen, die zijn opgesteld voor het onderzoek, zijn:

- Presteert een beslisboomalgoritme beter dan een handmatig opgesteld regelgebaseerd algoritme?

- Zijn er verschillen in correctheid tussen talen?

Binnen het vakgebied van kunstmatige intelligentie wordt er in dit onderzoek gefocust op het automatiseren van een menselijke taak door het toepassen van machine-learning op bepaalde datasets, in dit geval een deelverzameling van een corpus. De classificeerder moet nieuwe voorbeelden kunnen classificeren nadat die getraind is.

Theoretisch kader

Het automatisch toekennen van werkwoordstijden aan werkwoordscombinaties is een gebied dat nog niet erg bekend is. Hierdoor is er niet veel vergelijkbare literatuur beschikbaar waar dit onderzoek zich in kan kaderen. In deze sectie wordt meer informatie gegeven over de achtergrond van het onderzoek, de data en het algortime. Aan het einde zijn de hypotheses verwoord en onderbouwd

Zoals reeds al vermeld is, is het onderzoek onderdeel van het Time in Translation project. Binnen dit project zijn er al meerdere onderzoeken geweest. Een deel van deze onderzoeken maakt gebruikt van een bepaalde "tool"die is gemaakt door de hoofdonderzoekers van het project (van der Klis, Le Bruyn & de Swart, 2017). Deze tool heet de *Translation mining tool*. Met dit programma wordt er een semantische kaart gemaakt van hoe verschillende tijden worden gebruikt binnen een bepaald corpus in verschillende talen. Doordat er een duidelijke visualisatie is van hoe tijden zich verhouden in een corpus, valt er eenvoudiger een vergelijking te maken of een verschil te constateren tussen verschillende werkwoordstijden tussen twee of meer talen. Het proces van data tot de visualisatie bestaat uit 5 stappen:

- Extractie van zinnen met een daarin een voltooid werkwoord. Als in één zin een voltooid werkwoord voorkomt, worden de vertalingen van deze zin ook geëxtraheerd. Hierdoor ontstaat er een vijftal van zinnen waarin minimaal één keer een voltooid deelwoord voorkomt.

- Een annotator markeert de werkwoordscombinaties in deze vijf zinnen. De annotator kan ook foute vertalingen aangeven.

- Het toekennen van werkwoordstijden aan de werkwoordscombinaties. Voor het Nederlands, Engels en Frans is dit gedaan aan de hand van de Part of Speech tags. Voor het Spaans en het Duits is dit handmatig gedaan. De reden hiervoor was dat de tags voor het Spaans en het Duits niet duidelijk genoeg waren.

- Het bouwen van een matrix die uitdrukt hoe groot de verschillen zijn tussen de talen.

- Het visualiseren van de matrix uit stap vier door middel van een interface.

Aanvankelijk was er een poging gedaan om stap drie uit te voeren aan de hand van regelgebaseerde classificatie, maar deze bleek voornamelijk voor het Spaans niet adequate scores te behalen. Voor het Nederlands werd eerst het aantal werkwoorden in de zin geteld. Als er één werkwoord in de zin stond, werd direct afgeleid of de werkwoordstijd o.v.t. of o.t.t. was. Als er een hulpwerkwoord en een voltooid deelwoord in de zin stonden, werd er gecontroleerd of de werkwoordstijd de v.t.t. of de v.v.t. was. Werkwoordscombinaties die niet werden herkend, kregen 'other' toegekend, een soort restcategorie. Voor het Spaans werden er vergelijkbare stappen uitgevoerd, maar hiernaast werden ook nog de werkwoordsuitgangen gecontroleerd. Hiermee kan er onderscheid worden gemaakt tussen de 'pretérito indefinido' en de 'imperfecto'. Het Nederlands kent hier geen onderscheid in en heeft deze

extra stap niet nodig. Dit regelgebaseerde algoritme wordt gebruikt als basislijn voor het beslisboomalgoritme

Beschrijving inputdata

Er is gebruik gemaakt van twee verschillende datasets in dit onderzoek: De OpenSubtitles dataset en de Europarl dataset. De OpenSubtitles dataset is opgebouwd tijdens een ander bacheloronderzoek (Verkleij & Wimmers, 2016) en is gebaseerd op de data uit het corpus *OpenSubtitles2016*.(Lison & Tiedemann, 2016) In het bacheloronderzoek (Verkleij & Wimmers, 2016) is een goede beschrijving gegeven hoe de dataset er precies uitziet en hoe deze tot stand is gekomen. De Europarl dataset is opgezet tijdens een ander onderzoek(van der Klis, Le Bruyn & de Swart, 2017) en de data komt uit het *Europarl (v7)* (Tiedemann, 2012). In appendix A wordt nader beschreven wat voor werkwoordstijden er zijn per taal en hoe deze verdeeld zijn per dataset. De datasets die voor het onderzoek gebruikt zijn zullen op een later moment openbaar worden gemaakt op de website van het Time-in-Translation project.

De datasets bestaan uit verschillende variabelen, waarvan een deel van belang is voor het onderzoek en het andere deel niet relevant is. De relevante variabelen zijn **Pos** (Part of Speech) en **Lem** (Lemma). Een Part of Speech (PoS) tag is een afkorting die aan een woord kan worden gegeven wat de functie van het woord uitdrukt. Het lemma verwijst naar het infinitief van het werkwoord uit de data. In tabel 1 wordt dit nader uitgelegd aan de hand van de zinnen uit figuur 1. De PoS- en lemma tags zijn automatisch geannoteerd door een part-of-speech tagger.[1] Hoeveel PoS- en lemma tags er voorkomen hangt van de data af. Als er een zin uit de data is met bijvoorbeeld vier werkwoorden, betekent dat de er vier PoS tags en vier lemma tags zijn. De laatste variabele is **Target**. Dit is de werkelijke werkwoordstijd die bij de werkwoordscombinatie hoort. Deze wordt op huidig moment nog toegekend door annotatoren.

Ww1	Ww2	Lem1	Lem2	PoS1	PoS2
heeft	geslapen	hebben	slapen	vbpressg	vbpapa
liep		lopen		vbpastsg	
was	gekomen	zijn	komen	vbpastsg	vbpapa
loopt		lopen		vbpastsg	

Tabel 1: Dit zijn de werkwoorden van de zinnen uit figuur één. De PoS waarde duidt hier de functie van de werkwoorden aan. Zo betekenen 'vbpressg', 'vbpapa' en 'vbpastsg' respectievelijk 'verb present singular', 'verb past participle' en 'verb past singular'. Dit zijn een paar waarden voor het Nederlands. Voor andere talen worden er mogelijk andere afkortingen gebruikt.

[1]TreeTagger: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Het algoritme

Het algoritme is geschreven in Python 2.7. De broncode zal beschikbaar worden gesteld via Github[2]. Instructies voor het draaien van het algoritme staan beschreven in het bijgeleverde commentaar bij de code.

Voor het onderzoek wordt er gebruik gemaakt van een beslisboom. Een beslisboom bestaat uit knopen en bladeren. In de knopen wordt er aan de hand van een bepaald criterium een splitsing gemaakt. Hierdoor ontstaan er nieuwe knopen of bladeren. Een blad betekent dat er een concrete tijd wordt toegekend aan een bepaalde zin.

Er is besloten om gebruik te maken van een beslisboom om een zin te classificeren voor tweel redenen:

- Het is een white-box algoritme. Dit betekent dat het duidelijk te zien is hoe het algoritme een voorbeeld classificeert. Uit een boom valt af te lezen wat de criteria zijn en dus hoe een datavoorbeeld wordt geclassificeerd. Hierdoor vallen rariteiten in uitkomsten snel op.

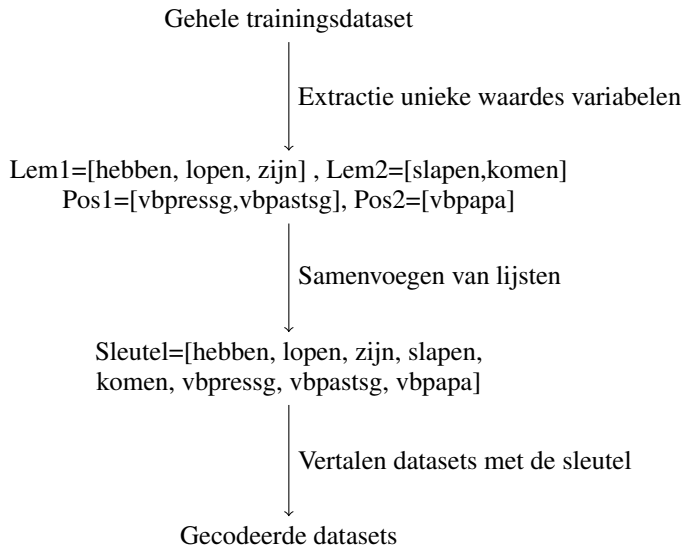
- Het functioneert op een vergelijkbare manier als het regelgebaseerde algoritme. Er wordt niet gezocht naar een optimale formule of functie zoals met regressie of naar de juiste gewichten voor een neuraal netwerk. Een regel uit het regelgebaseerde algoritme kan worden vertaald naar één of meerdere knopen met overeenkomende criteria van een boom.

Met de beslisboom wordt er een poging gedaan om patronen in de werkwoordscombinaties te herkennen die in de zin zijn gebruikt en uiteindelijk het voorbeeld te classificeren. Voor het classificeren wordt er gebruik gemaakt van een reeds bestaande beslisboom classificeerder. (Pedregosa F. et al. , 2011)[3] Deze classificeerder wordt in het algoritme geïncorporeerd. Om de data bruikbaar te maken voor de classificeerder, aangezien deze geen woorden als invoer neemt, wordt er een transformatie uitgevoerd van een aantal variabelen uit de dataset. Door middel van OneHot encodering worden alle variabelen getransformeerd naar een waarde van nul of één. Het gevolg hiervan is dat waar een trainingsvoorbeeld nu enkele woorden als waarden heeft , deze sterk kan oplopen afhankelijk van de hoeveelheid verschillende waarden van een variabele binnen een dataset. Aan de hand van een voorbeeld leg ik kort uit hoe OneHot encodering in z'n werking gaat. Voor een uitgebreidere uitleg verwijs ik naar het artikel van Vasudev (2017)[4].

[2]Link naar de pagina: https://github.com/time-in-translation/tense-assignment

[3]Link naar de gebruikte classificeerder: http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

[4]Link naar de uitleg: https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f



Figuur 3: Uitleg OneHot encodering aan de hand van de dataset uit tabel 2.

Stel dat de dataset waarop getraind wordt bestaat uit de vier zinnen uit figuur 1. Voordat een voorbeeld of een set data gecodeerd kan worden, moet de vertaalsleutel worden gemaakt. De vertaalsleutel is alleen gebaseerd op de trainingsdataset. De variabelen die gebruikt zullen worden voor het maken van de sleutel zijn: **Lemma1, Lemma2, Pos1, Pos2**. **Id** is voor het onderzoek niet relevant maar wordt gebruikt bij het voorbeeld om aan te tonen dat het verschillende zinnen zijn. **Target** is wel belangrijk bij het trainen van de boom, maar heeft geen waarde bij het encoderen van de data. Deze vier variabelen zijn uit de zinnen geëxtraheerd, waardoor niet de zinnen zelf als data worden gebruikt. Voor alle variabelen worden de unieke waarden gezocht uit de trainingsdata en worden deze in een lijst gestopt. In het voorbeeld zullen eerst de verschillende waarden van Lemma1 worden verzameld, vervolgens voor Lemma2, tot de laatste variabele. Deze lijst is de 'vertaalsleutel'. Aan de hand van deze sleutel kunnen data worden gecodeerd. Omdat **Lem1** drie verschillende elementen bevat, worden deze elementen gebruikt om de eerste drie binaire waarden vast te stellen van een datapunt. Als de waarde uit een datapunt gelijk is aan een van de elementen van **Lem1** komt op deze plek een één te staan en de andere twee plaatsen een nul. Voor het eerste voorbeeld uit tabel 2 wordt het lijstje van binaire waarden voor **Lem1** [1,0,0]. Voor de tweede variabele, **Lem2**, wordt dit [1,0]. Deze stap wordt vervolgens verder uitgevoerd voor de overige variabelen. Uiteindelijk ontstaan dus de lijstjes [1,0,0],[1,0],[1,0],[1]. Deze lijsten worden aanéén gevoegd. Het resultaat hiervan is het nieuwe gecodeerde datapunt, een lijst van binaire waarden. In figuur 4 zijn de gecodeerde datapunten weergegeven.

Onderbouwing deelvragen en hypothesen

De twee deelvragen zijn: 1. Presteert een beslisboomalgoritme beter dan een handmatig opgesteld regelgebaseerd algoritme? 2. Zijn er verschillen in correctheid tussen talen?

Id	Lem1	Lem2	Pos1	Pos2	Targ
1	Hebben	Slapen	vbpressg	vbpapa	vtt
2	Lopen		vbpastsg		ovt
3	Zijn	Komen	vbpastsg	vbpapa	vvt
4	Lopen		vbpressg		ott

Tabel 2: De ongecodeerde data bij de zinnen uit figuur 1. Deze zinnen komen uit de trainingsdata.

Zin 1: [1,0,0,1,0,1,0,1]
Zin 2: [0,1,0,0,0,0,1,0]
Zin 3: [0,0,1,0,1,0,1,1]
Zin 4: [0,1,0,0,0,1,0,0]

Figuur 4: Nieuwe datapunten na transformatie naar binaire waarden.

De eerste deelvraag is gesteld om een vergelijking mogelijk te maken tussen twee algoritmes. Omdat er nog nauwelijks onderzoek is gedaan naar het automatisch toekennen van tijden aan werkwoordscombinaties, zijn er niet veel resultaten die kunnen worden gebruikt als vergelijkingsmateriaal. Door het vergelijken van deze twee algoritmes is het toch mogelijk om enigzins inzicht te krijgen van de efficiëntie van het boomalgoritme. De hypothese van het onderzoek is dat het boomalgoritme minstens even goede scores kan behalen als het regelgebaseerde algoritme.

De tweede deelvraag is gesteld om te kijken of er met één algoritme verschillende scores ontstaan voor verschillende talen. Uit onderzoek(van der Klis, Le Bruyn, de Swart, 2017) is al naar voren gekomen dat er voor het Spaans en het Duits problemen waren met de classificatie. Dit wordt meegenomen in dit onderzoek. De hypothese is dat er verschillen in resultaten zijn tussen het Spaans en het Duits en de overige talen. Het wordt verwacht dat er geen verschillen zijn in correctheid van classificaties tussen het Nederlands, Frans en Engels.

De methode

Om te onderzoeken of er een effectieve manier is om automatisch tijden toe te kennen aan zinnen is het totaal opgedeeld in drie delen :

- 1. Het bouwen van beslisbomen voor het Nederlands, Spaans, Engels, Frans en Duits aan de hand van een beslisboomalgoritme en het valideren op de testdataset.
- 2. Het verschil onderzoeken tussen de beslisbomen die worden gemaakt voor verschillende talen.
- 3. Het verzamelen van data van het valideren en analyses uitvoeren op de resultaten van het boomalgoritme.

1. Het doel van dit werkstuk is om een deel van het gehele proces van het project te automatiseren. Op het huidige moment voeren annotatoren handmatig de werkwoordstijden in

	Voorspelling = X	**Voorspelling $\neq$ X**
Target = X	True positive (Tp)	False negative (Fn)
Target $\neq$ X	False positive (Fp)	True negative (Tn)

Tabel 3: De vier mogelijke situaties die kunnen ontstaan bij het voorspellen van een werkwoordstijd X met een algoritme.

de datasets in. Aangezien dit een repetitieve activiteit is waarbij mogelijk fouten kunnen worden gemaakt, wordt geprobeerd om deze stap door een algoritme te laten doen. De resultaten van het beslisboomalgoritme worden vergeleken met de resultaten van het regelgebaseerde algoritme.

De classificeerder wordt eerst getraind op de OpenSubtitles dataset, hierna wordt er gevalideerd op het Europarl dataset. Voor de meeste talen zijn deze datasets van ongeveer gelijke grootte. De variabele waarop wordt gevalideerd in dit onderzoek is de maximale grootte van de boom. In dit onderzoek zijn er een aantal waarden gekozen voor de maximale diepte van de boom. Deze waarden zijn: [Onbeperkt,11,9,7,5,3]. Deze waarden zijn gekozen om scores vast te stellen bij verschillende boomdieptes en in wat voor mate het algoritme overfit bij grotere bomen. Er is sprake van overfitting als een algoritme leert van de ruis in de data of te precies een fit zoekt die past bij de trainingsdata. Hierdoor gaat de score van de trainingsdata omhoog, maar generaliseert het slechter over nieuwe data.

In de classificeerder zit een kleine willekeurigheidsfactor. Bij elke splitsing van de boom wordt er deels willekeurig naar een criterium gezocht. Hierdoor kunnen er verschilllen ontstaan tussen twee bomen die getraind zijn op dezelfde dataset en de uitkomsten die zij voorspellen bij het valideren. Om een representatief beeld te krijgen wordt er 50 keer een boom getraind en wordt deze vervolgens gevalideerd.
2. Het tweede onderdeel van het werkstuk is om te bekijken wat de structuur is van de bomen die er worden gebouwd voor het Nederlands en het Spaans en hoe deze zich tot elkaar verhouden. Wordt er een vergelijkbare boom gebouwd bij meerdere talen of zijn er geheel verschillende bomen? Dit is meer ter observatie en ter interpretatie. Hieraan worden geen verdere conclusies verbonden.

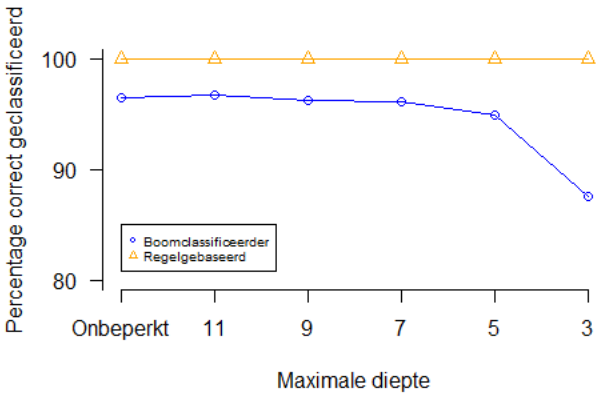
3. Het laatste onderdeel is om een analyse te maken van de resultaten van het algoritme. Om dit mogelijk te maken wordt er data verzameld. Bij iedere validatie wordt het aantal true positives, false positives en false negatives bijgehouden per werkwoordstijd. In tabel 3 is de betekenis van deze termen uitgelegd. Hieruit kunnen de gemiddelde percentages correcte classificaties worden berekend per tijd, per boomdiepte en globaal ook per taal. Het percentage correct geclassificeerd wordt berekend met de formule: $\frac{Tp}{Tp+Fn}$. In grafieken zullen deze percentages worden weergegeven. Vervolgens worden de f-scores berekend per boomdiepte met de volgende formule: $\frac{\frac{Tp}{Tp+Fp}+\frac{Tp}{Tp+Fn}}{2}$. De statistische testen zullen worden gebaseerd op de f-scores. De f-score is gekozen omdat het een completer beeld geeft dan het gemiddeld percentage correcte classificaties.

Evaluatie en analyse

In de sectie 'appendix B' zijn voor de vijf talen twee bomen weergegeven: Een boom met een diepte van r = 3 en r = 11. Deze bomen zijn geleverd om een indruk te geven hoe een gebouwde boom er uiteindelijk uit ziet. Verder worden deze bomen gebruikt om te zoeken naar overeenkomsten en verschillen tussen verschillende talen en dieptes. Bij verscheidene talen bestaat de eerste stap uit het controleren of de zin in een voltooide tijd staat of een voltooid deelwoord bevat. Zo wordt voor het Engels en het Spaans gecontroleerd of het eerste werkwoord een hulpwerkwoord is, waar voor het Nederlands en het Frans wordt gekeken naar de Part of Speech tag van het tweede werkwoord. Voor het Duits wordt er gecontroleerd of er een tweede werkwoord in de zin is. Voor alle talen wordt er dus een vergelijkbare eerste stap uitgevoerd. Na deze eerste stap worden er voor verschillende talen uiteenlopende bomen gebouwd. Een opvallend detail is dat de bomen met r = 3 geheel of grotendeels worden opgenomen in de bomen met r = 11. Hierdoor lijkt het dat een kleine boom de basis is waarop wordt gebouwd.

Analyse Nederlands en Spaans

De volgende resultaten zijn behaald bij het valideren op de Europarl dataset waarbij getraind is op de OpenSubtitles dataset.



Figuur 5: Het gemiddelde percentage correcte classificaties uitgezet per diepte van de boomclassificeerder voor het Nederlands. Het regelgebaseerde algoritme is toegevoegd als een basislijn.

In figuur 5 zijn de gemiddelde scores te zien voor het Nederlands bij verschillende boomdieptes. Per diepte is het gemiddelde percentage berekend uit de 50 validatiescores. De oranje lijn geeft de score weer van het regelgebaseerde algoritme. Voor alle dieptes behaalt het beslisboomalgoritme lagere scores dan het regelgebaseerde algoritme. De score van

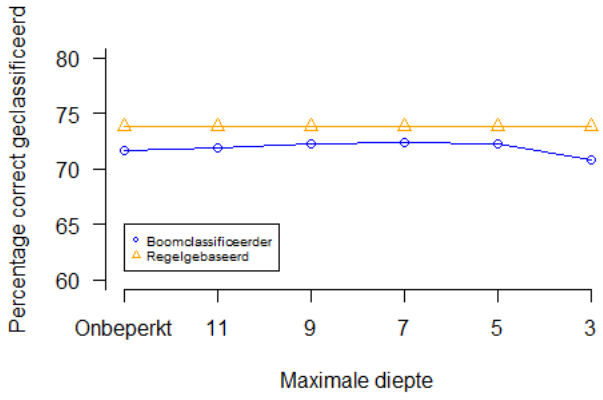
een boom met diepte 3 is lager dan de score van de grotere bomen.

De berekende f-scores staan in tabel 4. Aan de hand van de f-scores zijn er een aantal eenzijdige statistische t-toetsen uitgevoerd. Elke diepte wordt getoetst met de f-score van het regelgebaseerde algoritme. Deze heeft een waarde van 1.0 . Voor het onderzoek wordt een significantie α van 0.05 gehanteerd en zijn er 49 vrijheidsgraden omdat er 50 waarnemingen zijn. De f-score van het regelgebaseerde algoritme wordt als vergelijkingspunt gebruikt. De nulhypothese is $H_0 : \mu = 1.0$ en de alternatieve hypothese is $H_a : \mu < 1.0$.

Diepte	**F-score**	**Std. afwijking**	**P-waarde**
Onbeperkt	0.9650	0.009123	$2.2*10^{-16}$
11	0.9645	0.007748	$2.2*10^{-16}$
9	0.9647	0.009054	$2.2*10^{-16}$
7	0.9631	0.008669	$2.2*10^{-16}$
5	0.9487	0.007705	$2.2*10^{-16}$
3	0.8765	0.007525	$2.2*10^{-16}$

Tabel 4: De gemiddelde f-score per diepte, standaardafwijking en p-waarde voor het Nederlands

Voor alle dieptes geldt dat de p-waarde kleiner is dan α Hierdoor worden de nulhypothesen verworpen en de alternatieve hypothesen aangenomen. Statistisch gezien zijn er significante verschillen tussen de regelgebaseerde classificeerder en de boomclassificeerder voor het Nederlands.



Figuur 6: In deze grafiek zijn de percentages weergegeven per diepte. Het regelgebaseerde algoritme is weergegeven ter vergelijking.

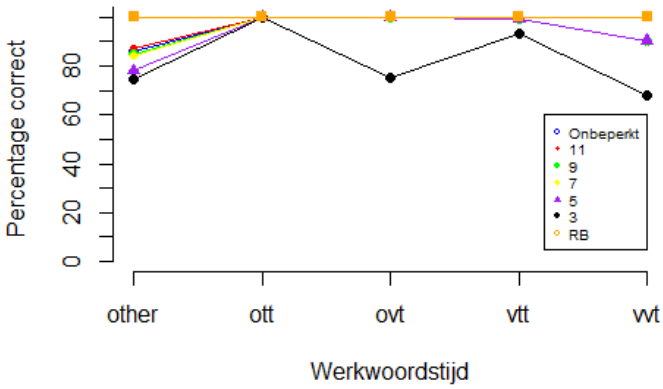
In figuur 6 worden de scores weergegeven van het regelgebaseerde algoritme en het beslisboomalgoritme voor het Spaans. Het verschil tussen de twee algoritmes is kleiner dan voor het Nederlands. Wederom zijn de scores van het beslisboomalgoritmes voor alle dieptes lager dan het regelgebaseerde algoritme. Bovendien zijn de percentages van de correcte classificaties aanzienlijk lager dan het Nederlands voor beide algoritmes. Beide algoritmes presteren ongeveer 25 procent slechter voor het Spaans dan het Nederlands. Voor het Spaans behaalde de regelgebaseerde classificeerder een f-score van 0.763. Voor statistische t-toetsen is voor het Spaans een α van 0.05 gebruikt en zijn er 49 vrijheidsgraden. In tabel 5 zijn de gemiddelde f-scores genoteerd van de boomclassificeerder. De f-score van elke diepte wordt getoetst met de waarde van het regelgebaseerde algoritme. De nulhypothese is $H_0 : \mu = 0.763$ en de alternatieve hypothese is $H_a : \mu < 0.763$.

Diepte	**F-score**	**Std. afwijking**	**P-waarde**
Onbeperkt	0.7171	0.002385	$2.2*10^{-16}$
11	0.7199	0.002532	$2.2*10^{-16}$
9	0.7235	0.004938	$2.2*10^{-16}$
7	0.7229	0.002588	$2.2*10^{-16}$
5	0.7217	0.002340	$2.2*10^{-16}$
3	0.7082	0.002340	$2.2*10^{-16}$

Tabel 5: In deze tabel zijn de gemiddelde f-scores genoteerd voor het Spaans bij de verschillende dieptes.

Uit de statistische toetsen zijn voor alle dieptes p-waarden behaald die kleiner zijn dan α. Hieruit blijken er significante verschillen te zijn tussen de uitkomsten van de twee algoritmes. Het beslisboomalgoritme presteert significant slechter dan het regelgebaseerde algoritme.

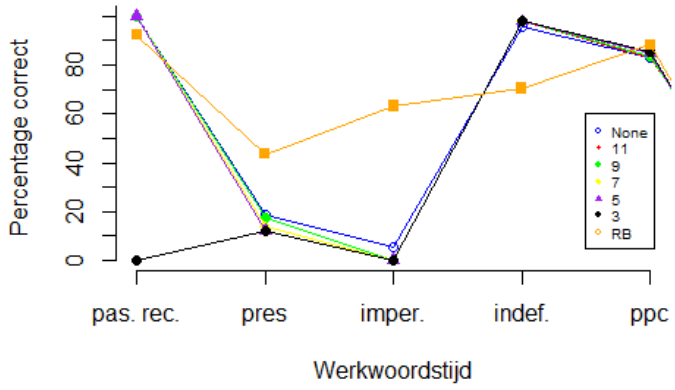
Beschrijvingen verschillen in dieptes



Figuur 7: Percentages correcte classificaties uitgezet voor verschillende boomdieptes per werkwoordscategorie voor het Nederlands. De oranje lijn geeft de prestaties van het regelgebaseerde algoritme weer. De 'other' categorie is een restcategorie van zinnen die niet geclassificeerd zijn onder één van de andere vier tijden.

In figuur 7 valt te zien dat het regelgebaseerde algoritme even goed of beter presteert dan het boomalgoritme. Er zijn

kleine verschillen tussen de verschillende dieptes. De scores van een boom met diepte van drie zijn lager bij de meeste tijden dan die van grotere bomen.



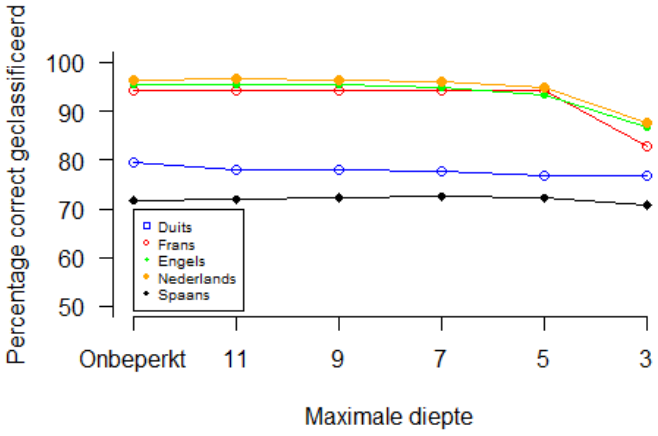
Figuur 8: Voor de vijf hoofdtijden zijn de prestaties genoteerd voor zowel verschillende boomdieptes als de regelgebaseerde classificeerder voor het Spaans. De overige tijden zijn weggelaten omdat deze niet werden herkend door beide classificeerders en percentages onder de 10 procent hadden.

Beide algoritmes functioneren matig voor het Spaans, wat te zien valt in figuur 8. Het regelgebaseerde algoritme classificeert de 'presente' benedenmaats. De boomclassificeerder classificeert zowel de 'presente' als de 'pretérito imperfecto' slecht. Tussen de boomdieptes onderling is niet veel verschil. Één uitschieter is de 'pasado reciente' bij boomdiepte drie. In appendix B valt te zien bij de Spaanse boom met een diepte van drie dat er geen enkel blad is waaraan de 'pasado reciente' als werkwoordstijd wordt toegekend. Dit verklaart waarom deze tijd niet kan worden toegekend bij deze diepte.

De overige talen

In figuur 9 is zijn de verschillende scores na het valideren te zien. De scores van het Frans, Nederlands en Engels liggen dicht bij elkaar. Het algoritme presteert aanzienlijk slechter voor het Duits en het Spaans ten opzichte van de overige talen. Voor het Nederlands, Engels, Frans en in kleine mate Spaans geldt dat er een val is in percentage bij een diepte van drie. Uit de f-scores in tabel 6 valt dit ook af te lezen.

Voor de vijf talen zijn uiteindelijk de f-scores verzameld en genoteerd in tabel 6. De scores van het Nederlands, Engels en Frans zijn hoger dan die van het Spaans en het Duits. Dit houdt in dat het algoritme accurater werkt voor deze talen dan voor het Spaans en Duits. Dit valt deels samen met eerder onderzoek (van der Klis, Le Bruyn, de Swart, 2017). In dit onderzoek is namelijk al vastgesteld dat de kwaliteit van de PoS tags niet goed genoeg was om werkwoordstijden toe te kennen aan de hand van deze tags. In dit onderzoek is vastgesteld dat het toevoegen van lemma tags hier geen verbetering in heeft aangebracht.



Figuur 9: In dit figuur zijn de percentages weergegeven voor alle vijf de talen.

diepte	nl	es	en	de	fr
None	**0.965**	0.7171	0.9539	**0.6834**	0.9427
11	0.9645	0.7199	**0.9543**	0.6708	**0.9429**
9	0.9647	0.**7235**	0.9539	0.6708	0.9422
7	0.9632	0.7229	0.949	0.6681	0.9424
5	0.9487	0.7217	0.9341	0.662	0.941
3	0.8765	0.7082	0.8687	0.6609	0.8277

Tabel 6: Bij de vijf talen zijn de f-scores genoteerd bij verschillende dieptes van bomen. Per taal is de beste score vetgedrukt.

In tabel 7 zijn de globale f-scores en standaardafwijkingen per taal genoteerd. Deze waardes zijn berekend door per taal te kijken naar de behaalde f-scores van alle dieptes. Per taal is dus gekeken naar 300 (6 maal 50) afzonderlijke waardes. Vervolgens is er steeds een tweetal van talen tegen elkaar getoetst met een t-toets om te controleren of deze significant verschillen. De geteste tweetallen zijn: Nederlands - Engels, Nederlands - Frans, Engels - Frans en Duits - Spaans. Er is een significantie α van 0.05 gehanteerd. De nulhypothese is: $H_0 : \mu_1 = \mu_2$. De alternatieve hypothese is $H_a : \mu_1 \neq \mu_2$.

Voor alle statistische testen zijn er p-waardes berekend die kleiner zijn dan α. Dit betekent dat het Nederlands, Engels en het Frans significant verschillen ten op zichte van elkaar. Het Spaans en het Duits verschillen ook significant van elkaar.

Taal	F-score	Std. afwijking
Nl	0.9471	0.03318
Es	0.7189	0.006042
Fr	0.9232	0.04277
En	0.9357	0.03109
De	0.6693	0.007643

Tabel 7: Globale f-scores van de vijf talen met bijbehorende standaardafwijkingen gebaseerd op 300 waarnemingen.

Conclusie

In dit onderzoek is er onderzocht of het mogelijk is om geautomatiseerd werkwoordstijden toe te kennen. De twee deelvragen hierbij waren: 1. Presteert een beslisboomalgoritme beter dan een handmatig opgesteld regelgebaseerd algoritme? 2. Zijn er verschillen in correctheid tussen talen? De hypothesen waren: Een beslisboomalgoritme presteert beter en er zijn geen verschillen in correctheid tussen talen. Uit de statistische toetsen is gebleken dat voor zowel het Nederlands als het Spaans het regelgebaseerde algoritme de beste resultaten opleverde. Hiernaast is ook gebleken door testen dat er wel degelijk verschillen zijn in correctheid tussen talen. Hierdoor zijn beide hypothesen onwaar en moeten verworpen worden. De terugkoppeling naar de hoofdvraag is dus dat het mogelijk is om geautomatiseerd werkwoordstijden toe te kennen aan werkwoordscombinaties, maar niet dat het noodzakelijk beter presteert dan een handmatig opgesteld algoritme en niet noodzakelijk voor alle talen even goed kan classificeren.

Het belangrijkste verbeterpunt dat uit dit onderzoek naar voren is gekomen is dat ik een gestructureerdere aanpak moet hanteren. In het begin van het onderzoek is er veel gedaan zonder dat er goed over is nagedacht. De keuze voor een beslisboom, bijvoorbeeld, was vroeg in het proces gemaakt, maar was niet goed onderbouwd en doordacht. Hierdoor was het moeilijk om aan goede literatuur te komen of een goede methode te ontwikkelen. Dit had als gevolg dat het lastig was een goede analyse uit te voeren.

Er zijn meerdere opties voor vervolgonderzoek. Er zou kunnen worden onderzocht of met andere algoritmes en data betere resultaten kunnen worden behaald dan in dit onderzoek voor een vergelijkbaar. Hiernaast zou het ook een interessant idee zijn om een vergelijkbaar onderzoek uit te voeren voor niet-westerse talen. Kunnen deze talen op een vergelijkbare manier worden behandeld als westerse talen, of is er juist een compleet nieuwe methode nodig?

Discussie

In dit onderzoek is er gekozen om een beslisboom te gebruiken om tijden toe te kennen aan werkwoordscombinaties. Hiernaast zijn er verschillende andere technieken die hiervoor kunnen worden ingezet. Een voorbeeld hiervan is een neuraal netwerk. Een vergelijkbaar onderzoek zou dus kunnen worden uitgevoerd, waarbij een neuraal netwerk wordt ingezet in plaats van een beslisboom. Vervolgens kan er worden gekeken of de ene structuur betere voorspellingen doet, beter bestand tegen ruis is of beter functioneert bij andere corpora of talen. Een beperking van een neuraal netwerk zou juist kunnen zijn dat er veel gevarieerde trainingsdata nodig is om het netwerk goed af te stellen. De data die gebruikt is voor dit onderzoek zou dan niet toereikend zijn.

Een nadeel van een beslisboom is dat het niet altijd het beste resultaat oplevert vergeleken met andere algoritmes. In (Curram & Mingers, 1994) is er een onderzoek uitgevoerd om de resultaten van drie algoritmes te vergelijken. Een neuraal netwerk, een boomclassificeerder en lineair discriminant analyse zijn gebruikt om voorbeelden uit een aantal verschillende datasets te classificeren. De boomclassificeerder behaalde globaal gezien de slechtste scores van deze drie methodes.

Het algoritme heeft voornamelijk bij het Spaans en Duits moeite om logische beslissingen te maken. Doordat de data, voornamelijk de Part of Speech tags, niet overal congruent is en voor verschillende tijden dezelfde tag gebruikt, is het niet altijd mogelijk om een goede splitsing te maken met de in het onderzoek gebruikte variabelen. Een mogelijke oplossing zou kunnen zijn om, bijvoorbeeld, werkwoordsverbuigingen af te vangen. Dit wordt ook toegepast in het regelgebaseerde algoritme, wat betere scores behaalt. Veel van de verschillende tijden in het Spaans hebben unieke uitgangen. Door het toevoegen van werkwoordsuitgang aan de data wordt misschien beslist op uitgang. Dit zou kunnen leiden tot een hogere nauwkeurigheid. Mogelijk is er een vergelijkbare oplossing voor het Duits.

target	w1	pos1	lemma1
imperfecto	tenías	VLfin	tener
indefinido	libramos	VLfin	librar
presente	reís	VLfin	reír

Tabel 8: datapunten uit de Spaanse trainingsset. Werkwoorden met verschillende werkwoordstijden hebben dezelfde PoS tag.

Literatuur

Curram, S. P. & Mingers, J. (1994). Neural networks, decision tree induction and discriminant analysis: an empirical comparison. *Journal of the Operational Research Society*, *45*, 440–450.

Lison, P. & Tiedemann, J. (2016, may). Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In N. C. C. Chair) et al. (red.), *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016).* Paris, France: European Language Resources Association (ELRA).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O.Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Tiedemann, J. (2012, May). Parallel data, tools and interfaces in opus. In N. Calzolari et al. (red.),

Proceedings of the eighth international conference on language resources and evaluation (lrec-2012) (pp. 2214–2218). Istanbul, Turkey: European Language Resources Association (ELRA). Verkrijgbaar van http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf (ACL Anthology Identifier: L12-1246)

van der Klis, M., Le Bruyn, B. & de Swart, H. (2017, April). Mapping the perfect via translation mining. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 2, short papers* (pp. 497–502). Valencia, Spain: Association for Computational Linguistics. Verkrijgbaar van http://www.aclweb.org/anthology/E17-2080

Vasudev, R. (2017 (bezocht Mei, 2018)). *One hot encoding.* Verkrijgbaar van https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f

Verkleij, A. & Wimmers, V. (2016). Filmperspectief op de present perfect. *Ongepubliceerde bachelorscriptie*.

Appendix A

In appendix A worden de verdelingen van werkwoordstijden van de data in de training- en validatiesets uitgebreider beschreven aan de hand van een aantal tabellen. In iedere tabel staat beschreven hoeveel datapunten er een bepaalde werkwoordstijd hebben in eeen bepaalde dataset. Met **train** wordt verwezen naar de OpenSubtitles dataset en met **validatie** naar de Europarl dataset. De tabellen laten goed zien dat niet iedere werkwoordstijd even sterk vertegenwoordigd is in een dataset.

tijd	**train**	**validatie**
condicional compuesto	14	0
condicional simple	4	1
futuro imperfecto	16	3
futuro perfecto	2	4
infinitivo	4	6
infinitivo compuesto	20	0
other	2	33
pasado reciente	8	26
presente	260	124
pretérito imperfecto	54	19
pretérito indefinido	836	196
pretérito perf. comp.	796	827
pretérito pluscuamperf.	34	11
participio	0	41

Tabel 9: De verdeling van werkwoordstijden per dataset voor het Spaans

tijd	**train**	**validatie**
cond. passé	6	2
cond. prés.	3	0
futur ant.	12	7
futur simp.	2	1
imparfait	23	13
other	95	153
passé comp.	44	940
passé récent	3	30
passé simple	1	0
plus-que parf.	6	18
présent	159	72

Tabel 10: Werkwoordstijden bij het Frans.

tijd	train	validatie
other	168	232
past cont.	5	3
past perfect	4	13
present cont.	11	2
present perf.	321	730
pres. perf. cont.	10	7
simple past	351	509
simple present	171	70

Tabel 11: Werkwoordstijden bij het Engels.

tijd	train	validatie
Futur I	4	3
Futur II	4	0
other	94	46
Perfekt	897	520
Plu. perf.	16	15
Präesens	74	214
Präteritum	344	109

Tabel 12: Werkwoordstijden bij het Duits.

tijd	train	validatie
other	272	294
ott	430	73
ovt	278	125
vtt	760	961
vvt	38	62

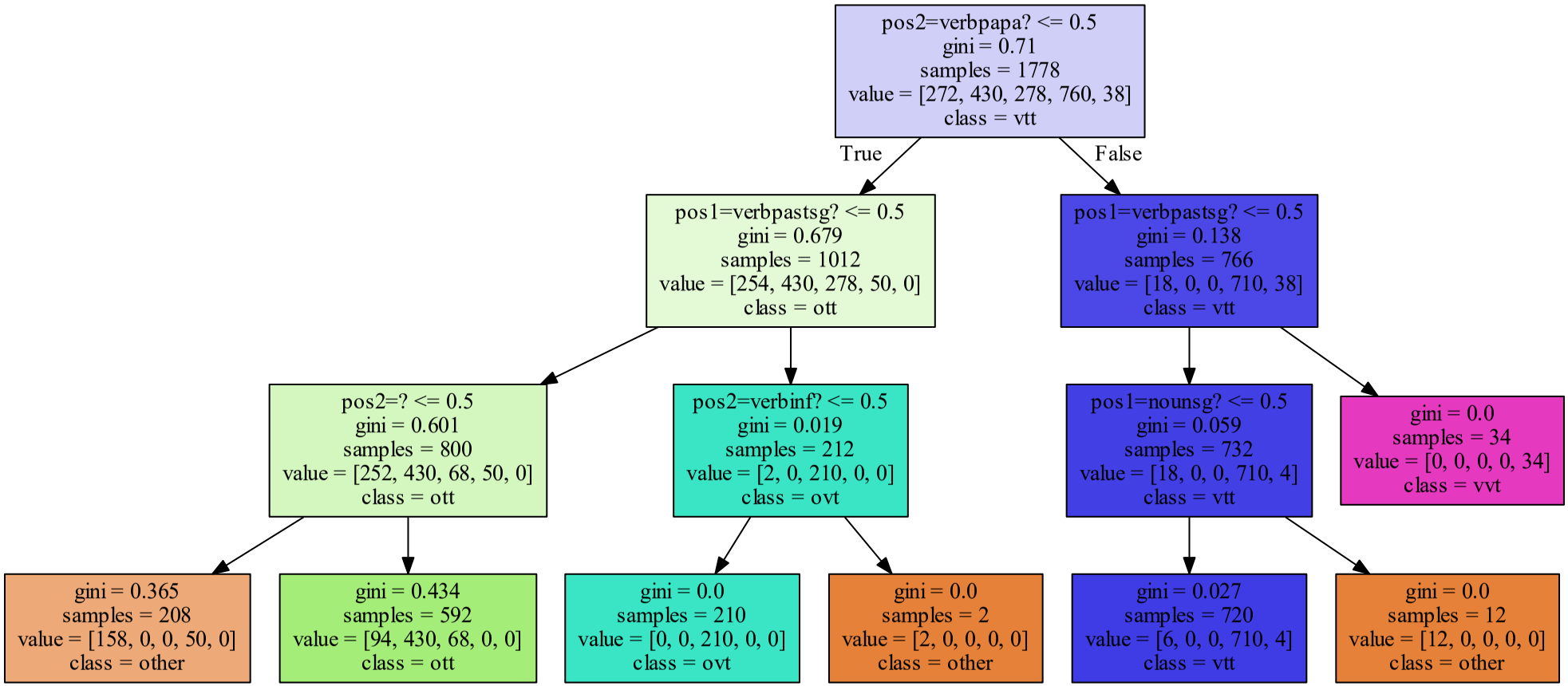
Tabel 13: Werkwoordstijden bij het Nederlands.

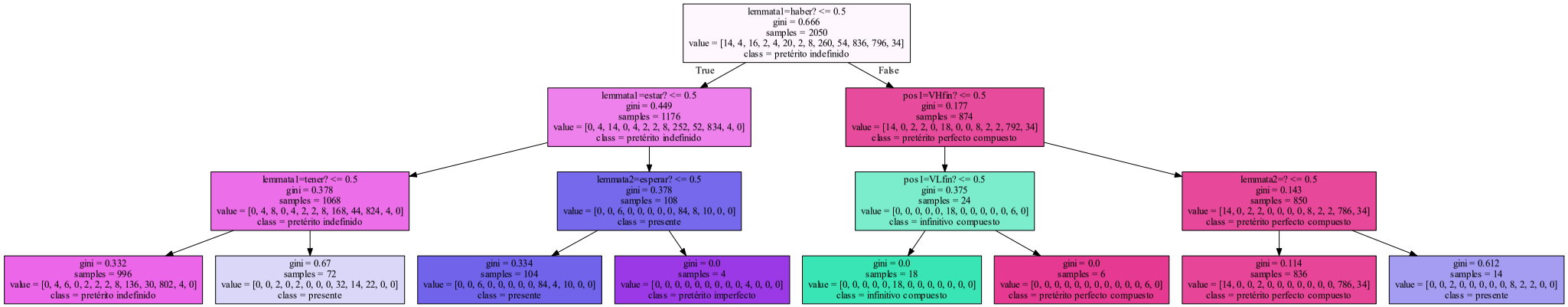
Appendix B

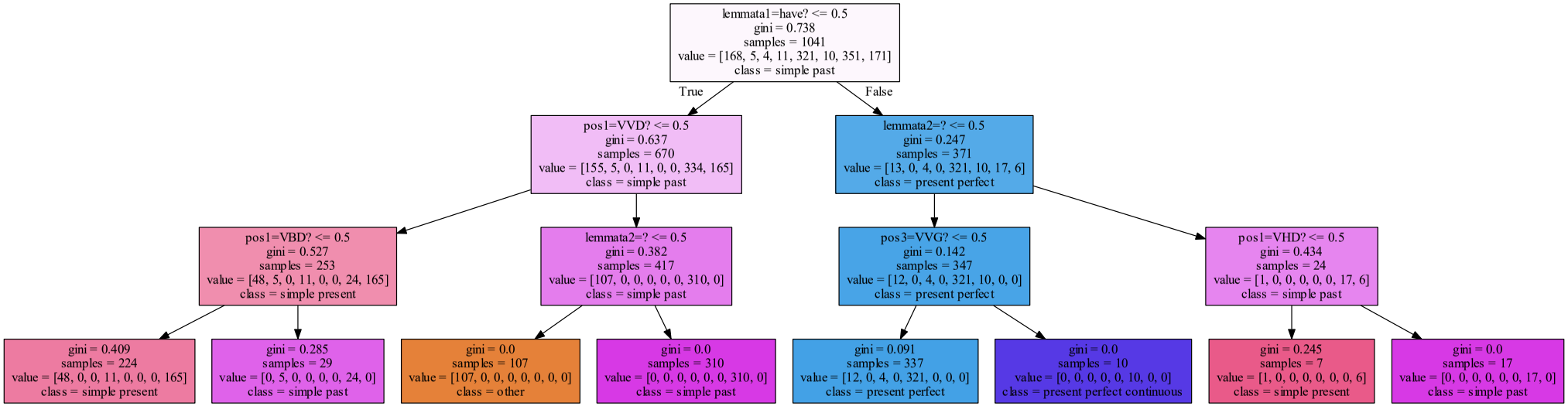
In de appendix B staan een aantal bomen die bij het onderzoek zijn gebouwd. In de boom valt te zien hoe trainingsvoorbeelden zijn gebruikt om een zo goed mogelijk functionerende boom te bouwen. In tabel 10 wordt de eerste knoop nader bekeken.

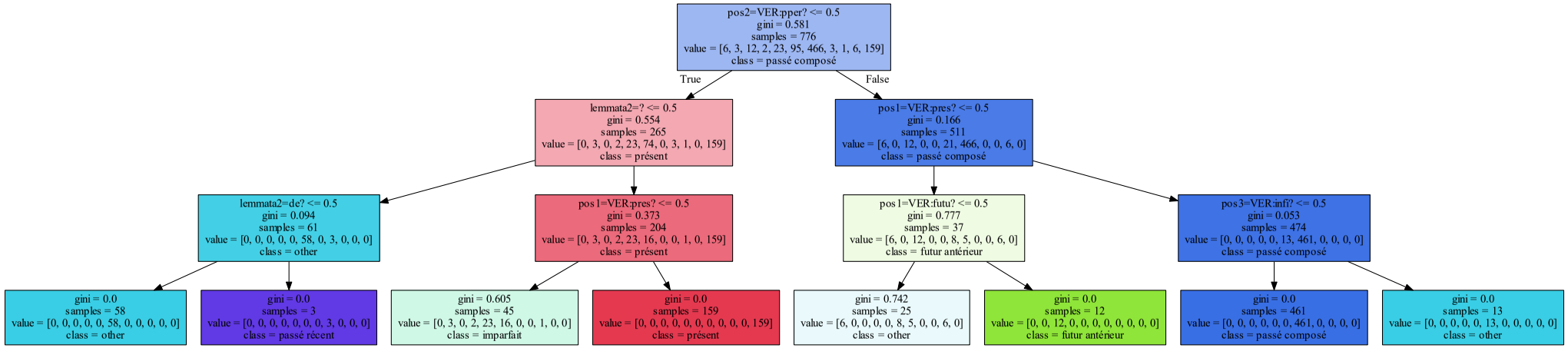
Statement	**Uitleg**
pos2=verbpapa = 0.5	Hier wordt gecontroleerd of PoS tag 2 van een voorbeeld 'verbpapa' is.
gini = 0.71	Dit is de waarde voor bepalen best mogelijke splitsing (alleen relevant bij het bouwen van de boom)
samples=1778	1778 voorbeelden worden gesplitst in deze knoop aan de hand van een criterium
value=[272,430,278,760,38]	Dit is de verdeling van de verschillende klassen die kunnen worden gesplitst.
Class = vvt	Werkwoordstijd die wordt toegekend aan voorbeelden in dit blad

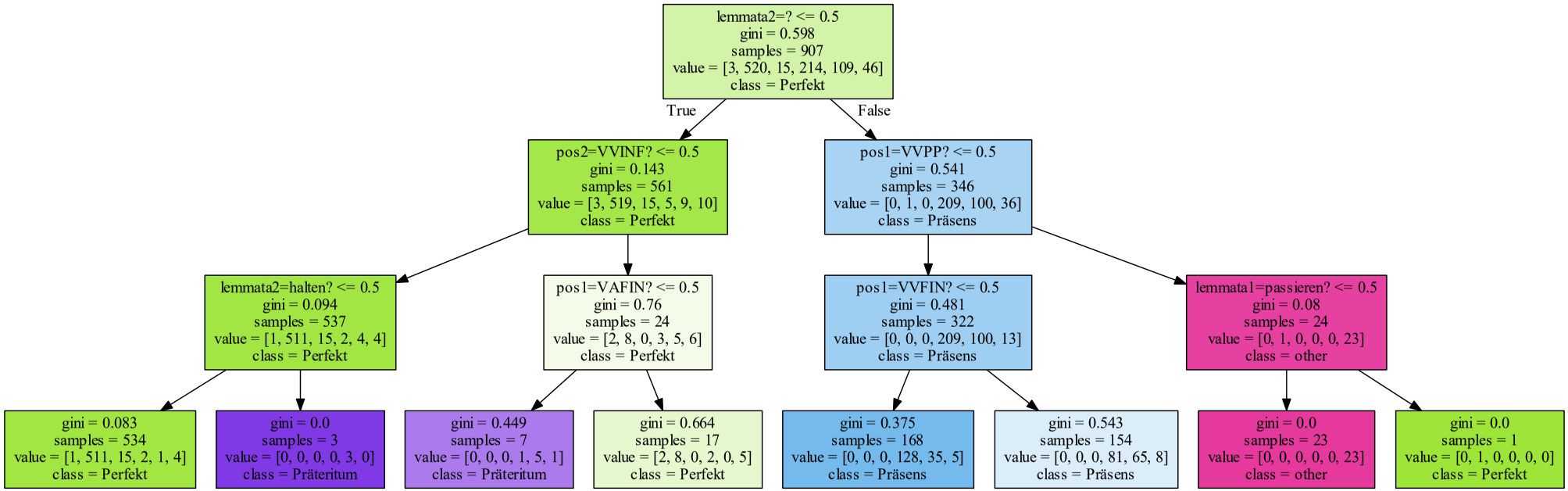
Tabel 14: Een beknopte uitleg van de informatie in de knopen en bladeren aan de hand van een voorbeeld.


pos2=verbpapa? <= 0.5
gini = 0.71
samples = 1778
value = [272, 430, 278, 760, 38]
class = vtt
True
False
pos1=verbpastsg? <= 0.5
gini = 0.679
samples = 1012
value = [254, 430, 278, 50, 0]
class = ott
pos1=verbpastsg? <= 0.5
gini = 0.138
samples = 766
value = [18, 0, 0, 710, 38]
class = vtt
pos2=? <= 0.5
gini = 0.601
samples = 800
value = [252, 430, 68, 50, 0]
class = ott
pos2=verbinf? <= 0.5
gini = 0.019
samples = 212
value = [2, 0, 210, 0, 0]
class = ovt
pos1=nounsg? <= 0.5
gini = 0.059
samples = 732
value = [18, 0, 0, 710, 4]
class = vtt
gini = 0.0
samples = 34
value = [0, 0, 0, 0, 34]
class = vvt
gini = 0.365
samples = 208
value = [158, 0, 0, 50, 0]
class = other
gini = 0.434
samples = 592
value = [94, 430, 68, 0, 0]
class = ott
gini = 0.0
samples = 210
value = [0, 0, 210, 0, 0]
class = ovt
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other
gini = 0.027
samples = 720
value = [6, 0, 0, 710, 4]
class = vtt
gini = 0.0
samples = 12
value = [12, 0, 0, 0, 0]
class = other

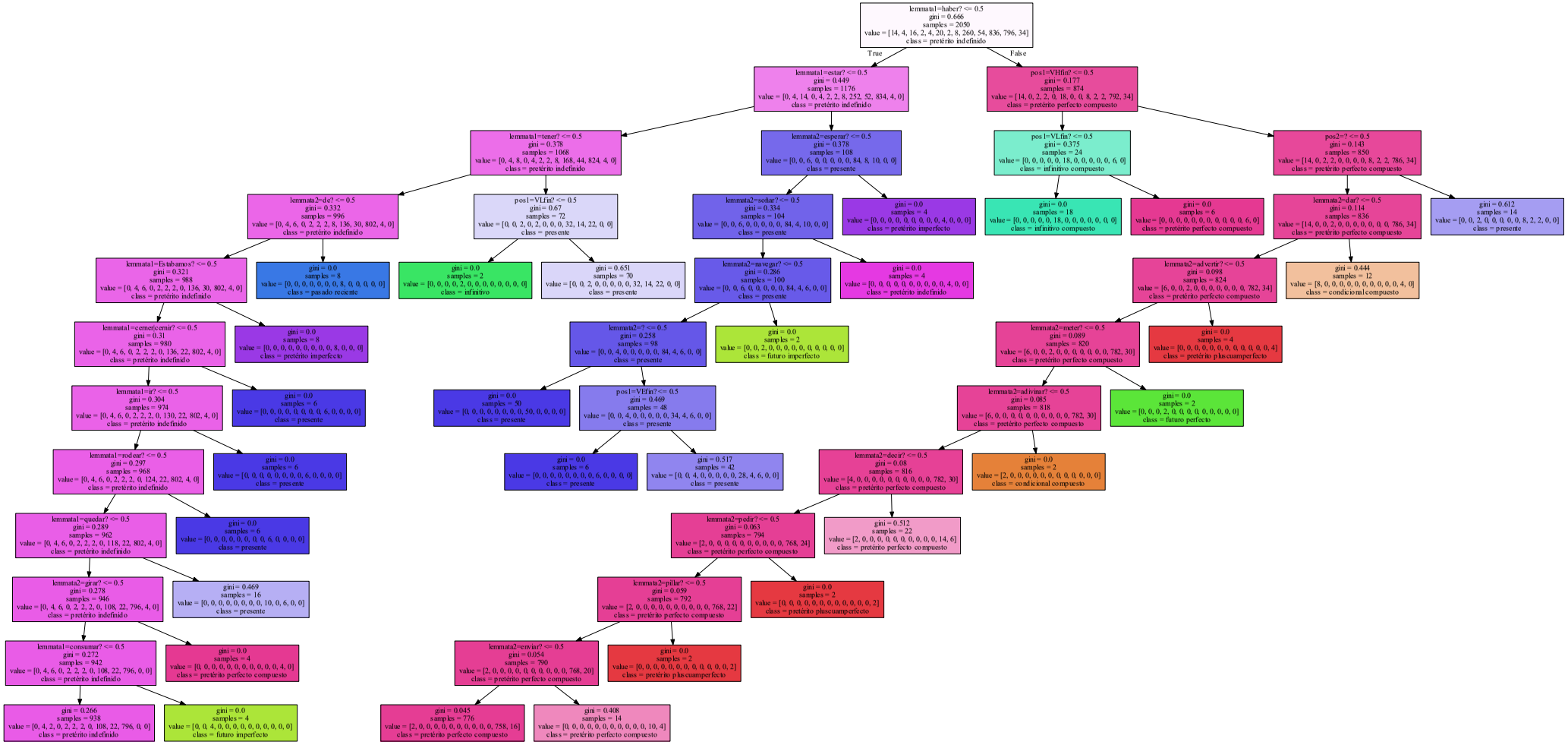

lemmata1=haber? <= 0.5
gini = 0.666
samples = 2050
value = [14, 4, 16, 2, 4, 20, 2, 8, 260, 54, 836, 796, 34]
class = pretérito indefinido
True
False
lemmata1=estar? <= 0.5
gini = 0.449
samples = 1176
value = [0, 4, 14, 0, 4, 2, 2, 8, 252, 52, 834, 4, 0]
class = pretérito indefinido
pos1=VHfin? <= 0.5
gini = 0.177
samples = 874
value = [14, 0, 2, 2, 0, 18, 0, 0, 8, 2, 2, 792, 34]
class = pretérito perfecto compuesto
lemmata1=tener? <= 0.5
gini = 0.378
samples = 1068
value = [0, 4, 8, 0, 4, 2, 2, 8, 168, 44, 824, 4, 0]
class = pretérito indefinido
lemmata2=esperar? <= 0.5
gini = 0.378
samples = 108
value = [0, 0, 6, 0, 0, 0, 0, 0, 84, 8, 10, 0, 0]
class = presente
pos1=VLfin? <= 0.5
gini = 0.375
samples = 24
value = [0, 0, 0, 0, 0, 18, 0, 0, 0, 0, 0, 6, 0]
class = infinitivo compuesto
lemmata2=? <= 0.5
gini = 0.143
samples = 850
value = [14, 0, 2, 2, 0, 0, 0, 0, 8, 2, 2, 786, 34]
class = pretérito perfecto compuesto
gini = 0.332
samples = 996
value = [0, 4, 6, 0, 2, 2, 2, 8, 136, 30, 802, 4, 0]
class = pretérito indefinido
gini = 0.67
samples = 72
value = [0, 0, 2, 0, 2, 0, 0, 0, 32, 14, 22, 0, 0]
class = presente
gini = 0.334
samples = 104
value = [0, 0, 6, 0, 0, 0, 0, 0, 84, 4, 10, 0, 0]
class = presente
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0]
class = pretérito imperfecto
gini = 0.0
samples = 18
value = [0, 0, 0, 0, 0, 18, 0, 0, 0, 0, 0, 0, 0]
class = infinitivo compuesto
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0]
class = pretérito perfecto compuesto
gini = 0.114
samples = 836
value = [14, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 786, 34]
class = pretérito perfecto compuesto
gini = 0.612
samples = 14
value = [0, 0, 2, 0, 0, 0, 0, 0, 8, 2, 2, 0, 0]
class = presente

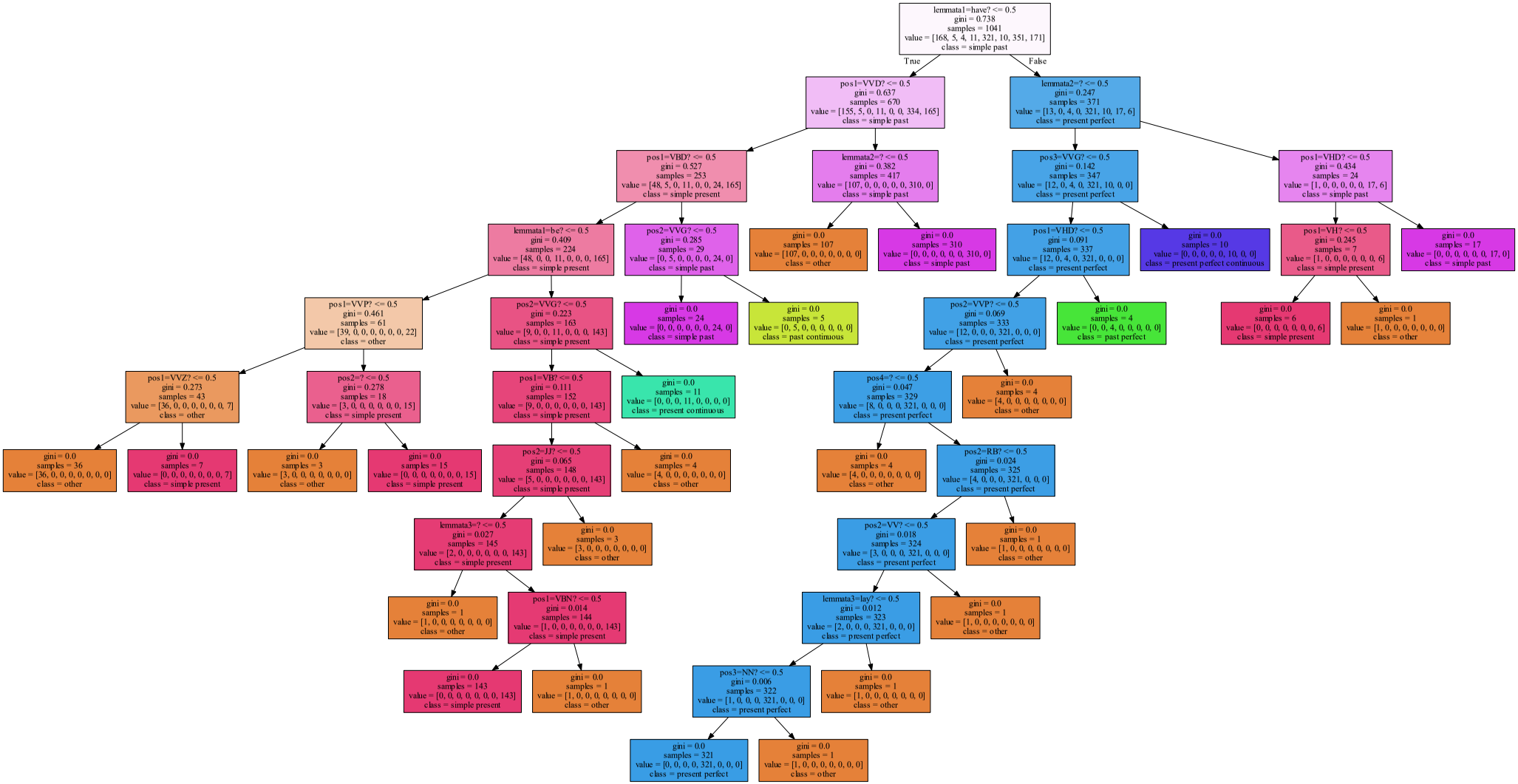

lemmata1=have? <= 0.5
gini = 0.738
samples = 1041
value = [168, 5, 4, 11, 321, 10, 351, 171]
class = simple past
True
False
pos1=VVD? <= 0.5
gini = 0.637
samples = 670
value = [155, 5, 0, 11, 0, 0, 334, 165]
class = simple past
lemmata2=? <= 0.5
gini = 0.247
samples = 371
value = [13, 0, 4, 0, 321, 10, 17, 6]
class = present perfect
pos1=VBD? <= 0.5
gini = 0.527
samples = 253
value = [48, 5, 0, 11, 0, 0, 24, 165]
class = simple present
lemmata2=? <= 0.5
gini = 0.382
samples = 417
value = [107, 0, 0, 0, 0, 0, 310, 0]
class = simple past
pos3=VVG? <= 0.5
gini = 0.142
samples = 347
value = [12, 0, 4, 0, 321, 10, 0, 0]
class = present perfect
pos1=VHD? <= 0.5
gini = 0.434
samples = 24
value = [1, 0, 0, 0, 0, 0, 17, 6]
class = simple past
gini = 0.409
samples = 224
value = [48, 0, 0, 11, 0, 0, 0, 165]
class = simple present
gini = 0.285
samples = 29
value = [0, 5, 0, 0, 0, 0, 24, 0]
class = simple past
gini = 0.0
samples = 107
value = [107, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 310
value = [0, 0, 0, 0, 0, 0, 310, 0]
class = simple past
gini = 0.091
samples = 337
value = [12, 0, 4, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 10
value = [0, 0, 0, 0, 0, 10, 0, 0]
class = present perfect continuous
gini = 0.245
samples = 7
value = [1, 0, 0, 0, 0, 0, 0, 6]
class = simple present
gini = 0.0
samples = 17
value = [0, 0, 0, 0, 0, 0, 17, 0]
class = simple past

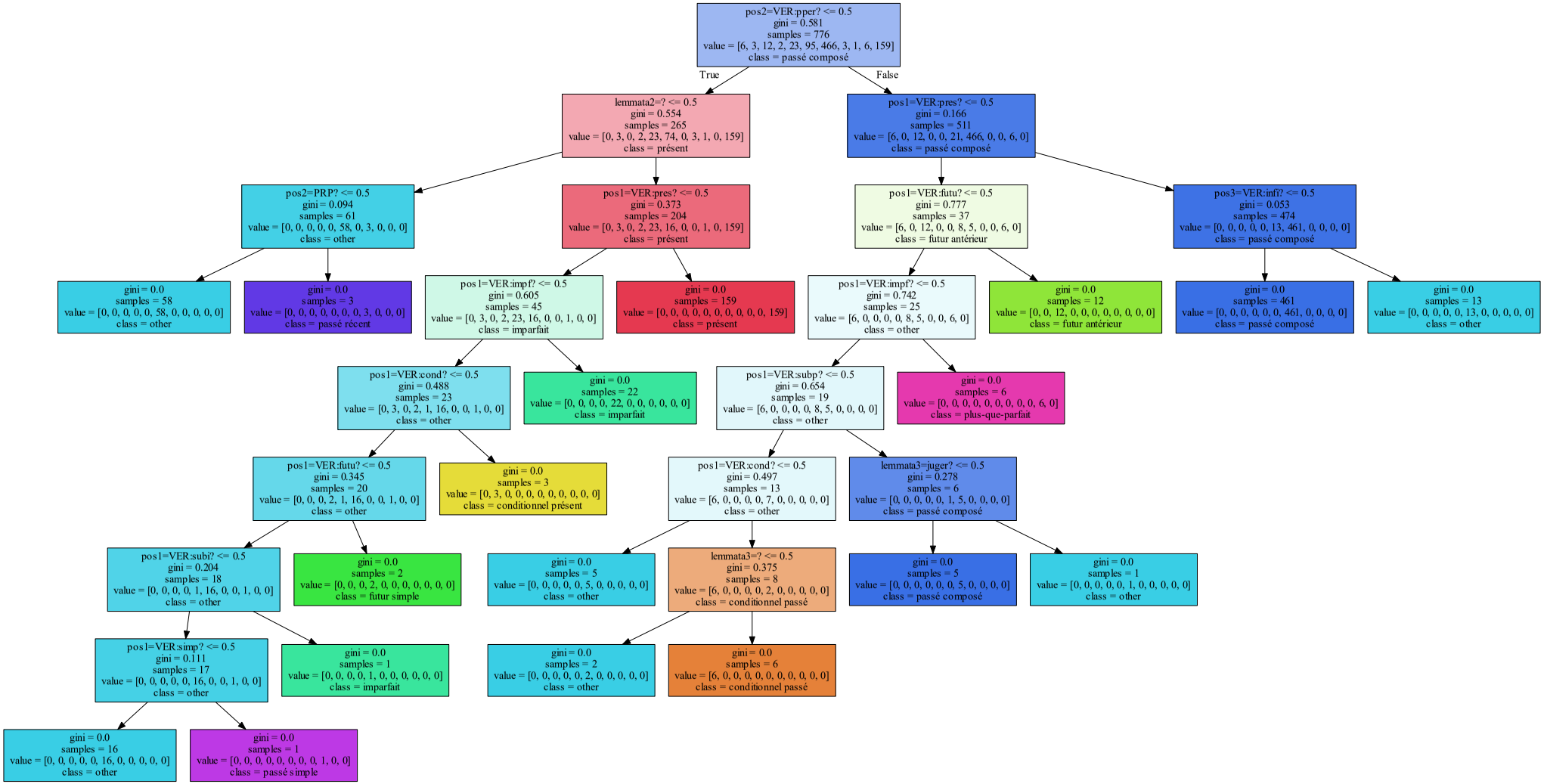

pos2=VER:pper? <= 0.5
gini = 0.581
samples = 776
value = [6, 3, 12, 2, 23, 95, 466, 3, 1, 6, 159]
class = passé composé
True
False
lemmata2=? <= 0.5
gini = 0.554
samples = 265
value = [0, 3, 0, 2, 23, 74, 0, 3, 1, 0, 159]
class = présent
pos1=VER:pres? <= 0.5
gini = 0.166
samples = 511
value = [6, 0, 12, 0, 0, 21, 466, 0, 0, 6, 0]
class = passé composé
lemmata2=de? <= 0.5
gini = 0.094
samples = 61
value = [0, 0, 0, 0, 0, 58, 0, 3, 0, 0, 0]
class = other
pos1=VER:pres? <= 0.5
gini = 0.373
samples = 204
value = [0, 3, 0, 2, 23, 16, 0, 0, 1, 0, 159]
class = présent
pos1=VER:futu? <= 0.5
gini = 0.777
samples = 37
value = [6, 0, 12, 0, 0, 8, 5, 0, 0, 6, 0]
class = futur antérieur
pos3=VER:infi? <= 0.5
gini = 0.053
samples = 474
value = [0, 0, 0, 0, 0, 13, 461, 0, 0, 0, 0]
class = passé composé
gini = 0.0
samples = 58
value = [0, 0, 0, 0, 0, 58, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 3
value = [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0]
class = passé récent
gini = 0.605
samples = 45
value = [0, 3, 0, 2, 23, 16, 0, 0, 1, 0, 0]
class = imparfait
gini = 0.0
samples = 159
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 159]
class = présent
gini = 0.742
samples = 25
value = [6, 0, 0, 0, 0, 8, 5, 0, 0, 6, 0]
class = other
gini = 0.0
samples = 12
value = [0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0]
class = futur antérieur
gini = 0.0
samples = 461
value = [0, 0, 0, 0, 0, 0, 461, 0, 0, 0, 0]
class = passé composé
gini = 0.0
samples = 13
value = [0, 0, 0, 0, 0, 13, 0, 0, 0, 0, 0]
class = other

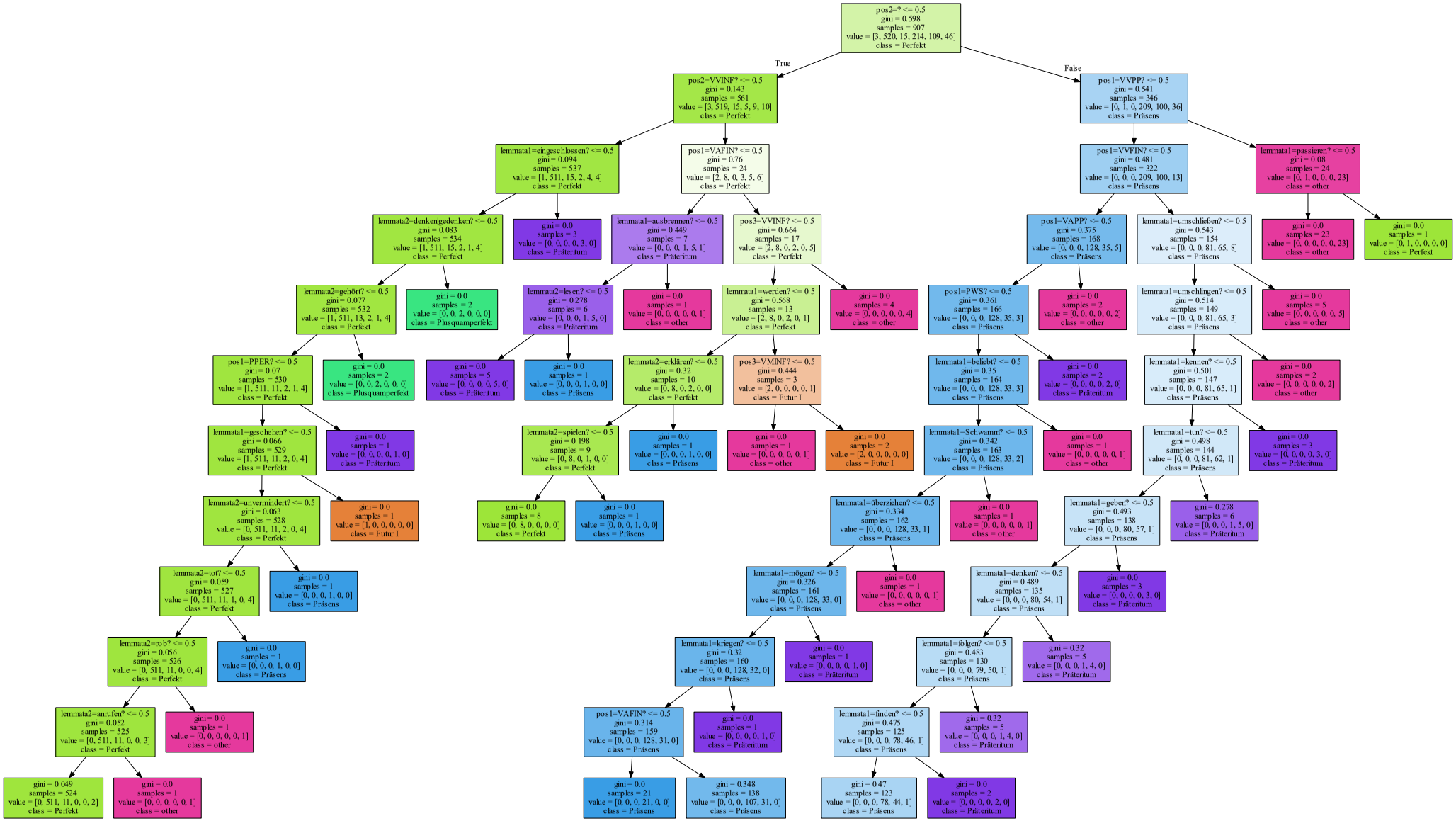

lemmata2=? <= 0.5
gini = 0.598
samples = 907
value = [3, 520, 15, 214, 109, 46]
class = Perfekt
True
False
pos2=VVINF? <= 0.5
gini = 0.143
samples = 561
value = [3, 519, 15, 5, 9, 10]
class = Perfekt
pos1=VVPP? <= 0.5
gini = 0.541
samples = 346
value = [0, 1, 0, 209, 100, 36]
class = Präsens
lemmata2=halten? <= 0.5
gini = 0.094
samples = 537
value = [1, 511, 15, 2, 4, 4]
class = Perfekt
pos1=VAFIN? <= 0.5
gini = 0.76
samples = 24
value = [2, 8, 0, 3, 5, 6]
class = Perfekt
pos1=VVFIN? <= 0.5
gini = 0.481
samples = 322
value = [0, 0, 0, 209, 100, 13]
class = Präsens
lemmata1=passieren? <= 0.5
gini = 0.08
samples = 24
value = [0, 1, 0, 0, 0, 23]
class = other
gini = 0.083
samples = 534
value = [1, 511, 15, 2, 1, 4]
class = Perfekt
gini = 0.0
samples = 3
value = [0, 0, 0, 0, 3, 0]
class = Präteritum
gini = 0.449
samples = 7
value = [0, 0, 0, 1, 5, 1]
class = Präteritum
gini = 0.664
samples = 17
value = [2, 8, 0, 2, 0, 5]
class = Perfekt
gini = 0.375
samples = 168
value = [0, 0, 0, 128, 35, 5]
class = Präsens
gini = 0.543
samples = 154
value = [0, 0, 0, 81, 65, 8]
class = Präsens
gini = 0.0
samples = 23
value = [0, 0, 0, 0, 0, 23]
class = other
gini = 0.0
samples = 1
value = [0, 1, 0, 0, 0, 0]
class = Perfekt


pos2=verbpapa? <= 0.5
gini = 0.71
samples = 1778
value = [272, 430, 278, 760, 38]
class = vtt
True
False
pos1=verbpastsg? <= 0.5
gini = 0.679
samples = 1012
value = [254, 430, 278, 50, 0]
class = ott
pos1=verbpastsg? <= 0.5
gini = 0.138
samples = 766
value = [18, 0, 0, 710, 38]
class = vtt
pos2=? <= 0.5
gini = 0.601
samples = 800
value = [252, 430, 68, 50, 0]
class = ott
lemmata2=luisteren? <= 0.5
gini = 0.019
samples = 212
value = [2, 0, 210, 0, 0]
class = ovt
pos1=nounsg? <= 0.5
gini = 0.059
samples = 732
value = [18, 0, 0, 710, 4]
class = vtt
gini = 0.0
samples = 34
value = [0, 0, 0, 0, 34]
class = vvt
pos1=verbpapa? <= 0.5
gini = 0.365
samples = 208
value = [158, 0, 0, 50, 0]
class = other
pos1=verbpastpl? <= 0.5
gini = 0.434
samples = 592
value = [94, 430, 68, 0, 0]
class = ott
gini = 0.0
samples = 210
value = [0, 0, 210, 0, 0]
class = ovt
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other
pos1=verbpastpl? <= 0.5
gini = 0.027
samples = 720
value = [6, 0, 0, 710, 4]
class = vtt
gini = 0.0
samples = 12
value = [12, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 158
value = [158, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 50
value = [0, 0, 0, 50, 0]
class = vtt
pos1=verbpapa? <= 0.5
gini = 0.294
samples = 524
value = [94, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 68
value = [0, 0, 68, 0, 0]
class = ovt
pos3=verbprespl? <= 0.5
gini = 0.017
samples = 716
value = [6, 0, 0, 710, 0]
class = vtt
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 4]
class = vvt
pos1=verbinf? <= 0.5
gini = 0.162
samples = 472
value = [42, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 52
value = [52, 0, 0, 0, 0]
class = other
lemmata2=snijden? <= 0.5
gini = 0.011
samples = 714
value = [4, 0, 0, 710, 0]
class = vtt
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other
pos1=nounsg? <= 0.5
gini = 0.093
samples = 452
value = [22, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 20
value = [20, 0, 0, 0, 0]
class = other
lemmata1=hed? <= 0.5
gini = 0.006
samples = 712
value = [2, 0, 0, 710, 0]
class = vtt
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other
lemmata1=komde? <= 0.5
gini = 0.044
samples = 440
value = [10, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 12
value = [12, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 710
value = [0, 0, 0, 710, 0]
class = vtt
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other
pos1=det__indef? <= 0.5
gini = 0.027
samples = 436
value = [6, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 4
value = [4, 0, 0, 0, 0]
class = other
pos1=det__poss? <= 0.5
gini = 0.018
samples = 434
value = [4, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other
lemmata1=inbraak? <= 0.5
gini = 0.009
samples = 432
value = [2, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 430
value = [0, 430, 0, 0, 0]
class = ott
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0]
class = other


lemmata1=haber? <= 0.5
gini = 0.666
samples = 2050
value = [14, 4, 16, 2, 4, 20, 2, 8, 260, 54, 836, 796, 34]
class = pretérito indefinido
True
False
lemmata1=estar? <= 0.5
gini = 0.449
samples = 1176
value = [0, 4, 14, 0, 4, 2, 2, 8, 252, 52, 834, 4, 0]
class = pretérito indefinido
pos1=VHfin? <= 0.5
gini = 0.177
samples = 874
value = [14, 0, 2, 2, 0, 18, 0, 0, 8, 2, 2, 792, 34]
class = pretérito perfecto compuesto
lemmata1=tener? <= 0.5
gini = 0.378
samples = 1068
value = [0, 4, 8, 0, 4, 2, 2, 8, 168, 44, 824, 4, 0]
class = pretérito indefinido
lemmata2=esperar? <= 0.5
gini = 0.378
samples = 108
value = [0, 0, 6, 0, 0, 0, 0, 0, 84, 8, 10, 0, 0]
class = presente
pos1=VLfin? <= 0.5
gini = 0.375
samples = 24
value = [0, 0, 0, 0, 0, 18, 0, 0, 0, 0, 0, 6, 0]
class = infinitivo compuesto
pos2=? <= 0.5
gini = 0.114
samples = 850
value = [14, 0, 2, 2, 0, 0, 0, 0, 8, 2, 2, 786, 34]
class = pretérito perfecto compuesto
lemmata2=de? <= 0.5
gini = 0.332
samples = 996
value = [0, 4, 6, 0, 2, 2, 2, 8, 136, 30, 802, 4, 0]
class = pretérito indefinido
pos1=VLfin? <= 0.5
gini = 0.67
samples = 72
value = [0, 0, 2, 0, 2, 0, 0, 0, 32, 14, 22, 0, 0]
class = presente
lemmata2=soñar? <= 0.5
gini = 0.334
samples = 104
value = [0, 0, 6, 0, 0, 0, 0, 0, 84, 4, 10, 0, 0]
class = presente
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0]
class = pretérito imperfecto
gini = 0.0
samples = 18
value = [0, 0, 0, 0, 0, 18, 0, 0, 0, 0, 0, 0]
class = infinitivo compuesto
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0]
class = pretérito perfecto compuesto
lemmata2=dar? <= 0.5
gini = 0.114
samples = 836
value = [14, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 786, 34]
class = pretérito perfecto compuesto
gini = 0.612
samples = 14
value = [0, 0, 2, 0, 0, 0, 0, 0, 0, 8, 2, 2, 0, 0]
class = presente
lemmata1=Estabamos? <= 0.5
gini = 0.321
samples = 988
value = [0, 4, 6, 0, 2, 2, 2, 0, 136, 30, 802, 4, 0]
class = pretérito indefinido
gini = 0.0
samples = 8
value = [0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 0]
class = pasado reciente
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0]
class = infinitivo
gini = 0.651
samples = 70
value = [0, 0, 2, 0, 0, 0, 0, 0, 32, 14, 22, 0, 0]
class = presente
lemmata2=navegar? <= 0.5
gini = 0.286
samples = 100
value = [0, 0, 6, 0, 0, 0, 0, 0, 84, 4, 6, 0, 0]
class = presente
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0]
class = pretérito indefinido
lemmata2=advertir? <= 0.5
gini = 0.098
samples = 824
value = [6, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 782, 34]
class = pretérito perfecto compuesto
gini = 0.444
samples = 12
value = [8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0]
class = condicional compuesto
lemmata1=cerrarcemir? <= 0.5
gini = 0.31
samples = 980
value = [0, 4, 6, 0, 2, 2, 2, 0, 136, 22, 802, 4, 0]
class = pretérito indefinido
gini = 0.0
samples = 8
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0]
class = pretérito imperfecto
lemmata2=?<= 0.5
gini = 0.258
samples = 98
value = [0, 0, 4, 0, 0, 0, 0, 0, 84, 4, 6, 0, 0]
class = presente
gini = 0.0
samples = 2
value = [0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = futuro imperfecto
lemmata2=meter? <= 0.5
gini = 0.089
samples = 820
value = [6, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 782, 30]
class = pretérito perfecto compuesto
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4]
class = pretérito pluscuamperfecto
lemmata1=ir? <= 0.5
gini = 0.304
samples = 974
value = [0, 4, 6, 0, 2, 2, 2, 0, 130, 22, 802, 4, 0]
class = pretérito indefinido
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0]
class = presente
gini = 0.0
samples = 50
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 50, 0, 0, 0]
class = presente
pos1=VEfin? <= 0.5
gini = 0.469
samples = 48
value = [0, 0, 4, 0, 0, 0, 0, 0, 34, 4, 6, 0, 0]
class = presente
lemmata2=adivinar? <= 0.5
gini = 0.085
samples = 818
value = [6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 782, 30]
class = pretérito perfecto compuesto
gini = 0.0
samples = 2
value = [0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = futuro perfecto
lemmata1=rodear? <= 0.5
gini = 0.297
samples = 968
value = [0, 4, 6, 0, 2, 2, 2, 0, 124, 22, 802, 4, 0]
class = pretérito indefinido
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0]
class = presente
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0]
class = presente
gini = 0.517
samples = 42
value = [0, 0, 4, 0, 0, 0, 0, 0, 28, 4, 6, 0, 0]
class = presente
lemmata2=decir? <= 0.5
gini = 0.08
samples = 816
value = [4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 782, 30]
class = pretérito perfecto compuesto
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = condicional compuesto
lemmata1=quedar? <= 0.5
gini = 0.289
samples = 962
value = [0, 4, 6, 0, 2, 2, 2, 0, 118, 22, 802, 4, 0]
class = pretérito indefinido
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0]
class = presente
lemmata2=pedir? <= 0.5
gini = 0.063
samples = 794
value = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 768, 24]
class = pretérito perfecto compuesto
gini = 0.512
samples = 22
value = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 6]
class = pretérito perfecto compuesto
lemmata1=grar? <= 0.5
gini = 0.278
samples = 946
value = [0, 4, 6, 0, 2, 2, 2, 0, 108, 22, 796, 4, 0]
class = pretérito indefinido
gini = 0.469
samples = 16
value = [0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 6, 0, 0]
class = presente
lemmata2=pillar? <= 0.5
gini = 0.059
samples = 792
value = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 768, 22]
class = pretérito perfecto compuesto
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
class = pretérito pluscuamperfecto
lemmata1=consumar? <= 0.5
gini = 0.272
samples = 942
value = [0, 4, 6, 0, 2, 2, 2, 0, 108, 22, 796, 0, 0]
class = pretérito indefinido
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0]
class = pretérito perfecto compuesto
lemmata2=enviar? <= 0.5
gini = 0.054
samples = 790
value = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 768, 20]
class = pretérito perfecto compuesto
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]
class = pretérito pluscuamperfecto
gini = 0.266
samples = 938
value = [0, 4, 2, 0, 2, 2, 2, 0, 108, 22, 796, 0, 0]
class = pretérito indefinido
gini = 0.0
samples = 4
value = [0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = futuro imperfecto
gini = 0.045
samples = 776
value = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 758, 16]
class = pretérito perfecto compuesto
gini = 0.408
samples = 14
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 4]
class = pretérito perfecto compuesto


lemmata1=have? <= 0.5
gini = 0.738
samples = 1041
value = [168, 5, 4, 11, 321, 10, 351, 171]
class = simple past
True
False
pos1=VVD? <= 0.5
gini = 0.637
samples = 670
value = [155, 5, 0, 11, 0, 0, 334, 165]
class = simple past
lemmata2=? <= 0.5
gini = 0.247
samples = 371
value = [13, 0, 4, 0, 321, 10, 17, 6]
class = present perfect
pos1=VBD? <= 0.5
gini = 0.527
samples = 253
value = [48, 5, 0, 11, 0, 0, 24, 165]
class = simple present
lemmata2=? <= 0.5
gini = 0.382
samples = 417
value = [107, 0, 0, 0, 0, 0, 310, 0]
class = simple past
pos3=VVG? <= 0.5
gini = 0.142
samples = 347
value = [12, 0, 4, 0, 321, 10, 0, 0]
class = present perfect
pos1=VHD? <= 0.5
gini = 0.434
samples = 24
value = [1, 0, 0, 0, 0, 0, 17, 6]
class = simple past
lemmata1=be? <= 0.5
gini = 0.409
samples = 224
value = [48, 0, 0, 11, 0, 0, 0, 165]
class = simple present
pos2=VVG? <= 0.5
gini = 0.285
samples = 29
value = [0, 5, 0, 0, 0, 0, 24, 0]
class = simple past
gini = 0.0
samples = 107
value = [107, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 310
value = [0, 0, 0, 0, 0, 0, 310, 0]
class = simple past
pos1=VHD? <= 0.5
gini = 0.091
samples = 337
value = [12, 0, 4, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 10
value = [0, 0, 0, 0, 0, 10, 0, 0]
class = present perfect continuous
pos1=VH? <= 0.5
gini = 0.245
samples = 7
value = [1, 0, 0, 0, 0, 0, 0, 6]
class = simple present
gini = 0.0
samples = 17
value = [0, 0, 0, 0, 0, 0, 17, 0]
class = simple past
pos1=VVP? <= 0.5
gini = 0.461
samples = 61
value = [39, 0, 0, 0, 0, 0, 0, 22]
class = other
pos2=VVG? <= 0.5
gini = 0.223
samples = 163
value = [9, 0, 0, 11, 0, 0, 0, 143]
class = simple present
gini = 0.0
samples = 24
value = [0, 0, 0, 0, 0, 0, 24, 0]
class = simple past
gini = 0.0
samples = 5
value = [0, 5, 0, 0, 0, 0, 0, 0]
class = past continuous
pos2=VVP? <= 0.5
gini = 0.069
samples = 333
value = [12, 0, 0, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 4
value = [0, 0, 4, 0, 0, 0, 0, 0]
class = past perfect
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 6]
class = simple present
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0, 0, 0]
class = other
pos1=VVZ? <= 0.5
gini = 0.273
samples = 43
value = [36, 0, 0, 0, 0, 0, 0, 7]
class = other
pos2=? <= 0.5
gini = 0.278
samples = 18
value = [3, 0, 0, 0, 0, 0, 0, 15]
class = simple present
pos1=VB? <= 0.5
gini = 0.111
samples = 152
value = [9, 0, 0, 0, 0, 0, 0, 143]
class = simple present
gini = 0.0
samples = 11
value = [0, 0, 0, 11, 0, 0, 0, 0]
class = present continuous
pos4=? <= 0.5
gini = 0.047
samples = 329
value = [8, 0, 0, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 4
value = [4, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 36
value = [36, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 7
value = [0, 0, 0, 0, 0, 0, 0, 7]
class = simple present
gini = 0.0
samples = 3
value = [3, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 15
value = [0, 0, 0, 0, 0, 0, 0, 15]
class = simple present
pos2=JJ? <= 0.5
gini = 0.065
samples = 148
value = [5, 0, 0, 0, 0, 0, 0, 143]
class = simple present
gini = 0.0
samples = 4
value = [4, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 4
value = [4, 0, 0, 0, 0, 0, 0, 0]
class = other
pos2=RB? <= 0.5
gini = 0.024
samples = 325
value = [4, 0, 0, 0, 321, 0, 0, 0]
class = present perfect
lemmata3=? <= 0.5
gini = 0.027
samples = 145
value = [2, 0, 0, 0, 0, 0, 0, 143]
class = simple present
gini = 0.0
samples = 3
value = [3, 0, 0, 0, 0, 0, 0, 0]
class = other
pos2=VV? <= 0.5
gini = 0.018
samples = 324
value = [3, 0, 0, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0, 0, 0]
class = other
pos1=VBN? <= 0.5
gini = 0.014
samples = 144
value = [1, 0, 0, 0, 0, 0, 0, 143]
class = simple present
lemmata3=lay? <= 0.5
gini = 0.012
samples = 323
value = [2, 0, 0, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 143
value = [0, 0, 0, 0, 0, 0, 0, 143]
class = simple present
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0, 0, 0]
class = other
pos3=NN? <= 0.5
gini = 0.006
samples = 322
value = [1, 0, 0, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 321
value = [0, 0, 0, 0, 321, 0, 0, 0]
class = present perfect
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0, 0, 0]
class = other


pos2=VER:pper? <= 0.5
gini = 0.581
samples = 776
value = [6, 3, 12, 2, 23, 95, 466, 3, 1, 6, 159]
class = passé composé
True
False
lemmata2=? <= 0.5
gini = 0.554
samples = 265
value = [0, 3, 0, 2, 23, 74, 0, 3, 1, 0, 159]
class = présent
pos1=VER:pres? <= 0.5
gini = 0.166
samples = 511
value = [6, 0, 12, 0, 0, 21, 466, 0, 0, 6, 0]
class = passé composé
pos2=PRP? <= 0.5
gini = 0.094
samples = 61
value = [0, 0, 0, 0, 0, 0, 58, 0, 3, 0, 0, 0]
class = other
pos1=VER:pres? <= 0.5
gini = 0.373
samples = 204
value = [0, 3, 0, 2, 23, 16, 0, 0, 1, 0, 159]
class = présent
pos1=VER:futu? <= 0.5
gini = 0.777
samples = 37
value = [6, 0, 12, 0, 0, 8, 5, 0, 0, 6, 0]
class = futur antérieur
pos3=VER:infi? <= 0.5
gini = 0.053
samples = 474
value = [0, 0, 0, 0, 0, 13, 461, 0, 0, 0, 0]
class = passé composé
gini = 0.0
samples = 58
value = [0, 0, 0, 0, 0, 58, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 3
value = [0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0]
class = passé récent
pos1=VER:impf? <= 0.5
gini = 0.605
samples = 45
value = [0, 3, 0, 2, 23, 16, 0, 0, 1, 0, 0]
class = imparfait
gini = 0.0
samples = 159
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 159]
class = présent
pos1=VER:impf? <= 0.5
gini = 0.742
samples = 25
value = [6, 0, 0, 0, 0, 8, 5, 0, 0, 6, 0]
class = other
gini = 0.0
samples = 12
value = [0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0]
class = futur antérieur
gini = 0.0
samples = 461
value = [0, 0, 0, 0, 0, 0, 461, 0, 0, 0, 0]
class = passé composé
gini = 0.0
samples = 13
value = [0, 0, 0, 0, 0, 13, 0, 0, 0, 0, 0]
class = other
pos1=VER:cond? <= 0.5
gini = 0.488
samples = 23
value = [0, 3, 0, 2, 1, 16, 0, 0, 1, 0, 0]
class = other
gini = 0.0
samples = 22
value = [0, 0, 0, 0, 22, 0, 0, 0, 0, 0, 0]
class = imparfait
pos1=VER:subp? <= 0.5
gini = 0.654
samples = 19
value = [6, 0, 0, 0, 0, 8, 5, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 6
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0]
class = plus-que-parfait
pos1=VER:futu? <= 0.5
gini = 0.345
samples = 20
value = [0, 0, 0, 2, 1, 16, 0, 0, 1, 0, 0]
class = other
gini = 0.0
samples = 3
value = [0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = conditionnel présent
pos1=VER:cond? <= 0.5
gini = 0.497
samples = 13
value = [6, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0]
class = other
lemmata3=juger? <= 0.5
gini = 0.278
samples = 6
value = [0, 0, 0, 0, 0, 1, 5, 0, 0, 0, 0]
class = passé composé
pos1=VER:subi? <= 0.5
gini = 0.204
samples = 18
value = [0, 0, 0, 0, 1, 16, 0, 0, 1, 0, 0]
class = other
gini = 0.0
samples = 2
value = [0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0]
class = futur simple
gini = 0.0
samples = 5
value = [0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0]
class = other
lemmata3=? <= 0.5
gini = 0.375
samples = 8
value = [6, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0]
class = conditionnel passé
gini = 0.0
samples = 5
value = [0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0]
class = passé composé
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
class = other
pos1=VER:simp? <= 0.5
gini = 0.111
samples = 17
value = [0, 0, 0, 0, 0, 16, 0, 0, 1, 0, 0]
class = other
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
class = imparfait
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 6
value = [6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = conditionnel passé
gini = 0.0
samples = 16
value = [0, 0, 0, 0, 0, 16, 0, 0, 0, 0, 0]
class = other
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
class = passé simple


pos2=? <= 0.5
gini = 0.598
samples = 907
value = [3, 520, 15, 214, 109, 46]
class = Perfekt
True
False
pos2=VVINF? <= 0.5
gini = 0.143
samples = 561
value = [3, 519, 15, 5, 9, 10]
class = Perfekt
pos1=VVPP? <= 0.5
gini = 0.541
samples = 346
value = [0, 1, 0, 209, 100, 36]
class = Präsens
lemmata1=eingeschlossen? <= 0.5
gini = 0.094
samples = 537
value = [1, 511, 15, 2, 4, 4]
class = Perfekt
pos1=VAFIN? <= 0.5
gini = 0.76
samples = 24
value = [2, 8, 0, 3, 5, 6]
class = Perfekt
pos1=VVFIN? <= 0.5
gini = 0.481
samples = 322
value = [0, 0, 0, 209, 100, 13]
class = Präsens
lemmata1=passieren? <= 0.5
gini = 0.08
samples = 24
value = [0, 1, 0, 0, 0, 23]
class = other
lemmata2=denken|gedenken? <= 0.5
gini = 0.083
samples = 534
value = [1, 511, 15, 2, 1, 4]
class = Perfekt
gini = 0.0
samples = 3
value = [0, 0, 0, 0, 3, 0]
class = Präteritum
lemmata1=ausbrennen? <= 0.5
gini = 0.449
samples = 7
value = [0, 0, 0, 1, 5, 1]
class = Präteritum
pos3=VVINF? <= 0.5
gini = 0.664
samples = 17
value = [2, 8, 0, 2, 0, 5]
class = Perfekt
pos1=VAPP? <= 0.5
gini = 0.375
samples = 168
value = [0, 0, 0, 128, 35, 5]
class = Präsens
lemmata1=umschließen? <= 0.5
gini = 0.543
samples = 154
value = [0, 0, 0, 81, 65, 8]
class = Präsens
gini = 0.0
samples = 23
value = [0, 0, 0, 0, 0, 23]
class = other
gini = 0.0
samples = 1
value = [0, 1, 0, 0, 0, 0]
class = Perfekt
lemmata2=gehört? <= 0.5
gini = 0.077
samples = 532
value = [1, 511, 13, 2, 1, 4]
class = Perfekt
gini = 0.0
samples = 2
value = [0, 0, 2, 0, 0, 0]
class = Plusquamperfekt
lemmata2=lesen? <= 0.5
gini = 0.278
samples = 6
value = [0, 0, 0, 1, 5, 0]
class = Präteritum
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1]
class = other
lemmata1=werden? <= 0.5
gini = 0.568
samples = 13
value = [2, 8, 0, 2, 0, 1]
class = Perfekt
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 4]
class = other
pos1=PWS? <= 0.5
gini = 0.361
samples = 166
value = [0, 0, 0, 128, 35, 3]
class = Präsens
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 0, 2]
class = other
lemmata1=umschlingen? <= 0.5
gini = 0.514
samples = 149
value = [0, 0, 0, 81, 65, 3]
class = Präsens
gini = 0.0
samples = 5
value = [0, 0, 0, 0, 0, 5]
class = other
pos1=PPER? <= 0.5
gini = 0.07
samples = 530
value = [1, 511, 11, 2, 1, 4]
class = Perfekt
gini = 0.0
samples = 2
value = [0, 0, 2, 0, 0, 0]
class = Plusquamperfekt
gini = 0.0
samples = 5
value = [0, 0, 0, 0, 5, 0]
class = Präteritum
gini = 0.0
samples = 1
value = [0, 0, 0, 1, 0, 0]
class = Präsens
lemmata2=erklären? <= 0.5
gini = 0.32
samples = 10
value = [0, 8, 0, 2, 0, 0]
class = Perfekt
pos3=VMINF? <= 0.5
gini = 0.444
samples = 3
value = [2, 0, 0, 0, 0, 1]
class = Futur I
lemmata1=beliebt? <= 0.5
gini = 0.35
samples = 164
value = [0, 0, 0, 128, 33, 3]
class = Präsens
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 2, 0]
class = Präteritum
lemmata1=kennen? <= 0.5
gini = 0.501
samples = 147
value = [0, 0, 0, 81, 65, 1]
class = Präsens
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 0, 2]
class = other
lemmata1=geschehen? <= 0.5
gini = 0.066
samples = 529
value = [1, 511, 11, 2, 0, 4]
class = Perfekt
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 1, 0]
class = Präteritum
lemmata2=spielen? <= 0.5
gini = 0.198
samples = 9
value = [0, 8, 0, 1, 0, 0]
class = Perfekt
gini = 0.0
samples = 1
value = [0, 0, 0, 1, 0, 0]
class = Präsens
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1]
class = other
gini = 0.0
samples = 2
value = [2, 0, 0, 0, 0, 0]
class = Futur I
lemmata1=Schwamm? <= 0.5
gini = 0.342
samples = 163
value = [0, 0, 0, 128, 33, 2]
class = Präsens
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1]
class = other
lemmata1=tun? <= 0.5
gini = 0.498
samples = 144
value = [0, 0, 0, 81, 62, 1]
class = Präsens
gini = 0.0
samples = 3
value = [0, 0, 0, 0, 3, 0]
class = Präteritum
lemmata2=unvermindert? <= 0.5
gini = 0.063
samples = 528
value = [0, 511, 11, 2, 0, 4]
class = Perfekt
gini = 0.0
samples = 1
value = [1, 0, 0, 0, 0, 0]
class = Futur I
gini = 0.0
samples = 8
value = [0, 8, 0, 0, 0, 0]
class = Perfekt
gini = 0.0
samples = 1
value = [0, 0, 0, 1, 0, 0]
class = Präsens
lemmata1=überziehen? <= 0.5
gini = 0.334
samples = 162
value = [0, 0, 0, 128, 33, 1]
class = Präsens
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1]
class = other
lemmata1=geben? <= 0.5
gini = 0.493
samples = 138
value = [0, 0, 0, 80, 57, 1]
class = Präsens
gini = 0.278
samples = 6
value = [0, 0, 0, 1, 5, 0]
class = Präteritum
lemmata2=tot? <= 0.5
gini = 0.059
samples = 527
value = [0, 511, 11, 1, 0, 4]
class = Perfekt
gini = 0.0
samples = 1
value = [0, 0, 0, 1, 0, 0]
class = Präsens
lemmata1=mögen? <= 0.5
gini = 0.326
samples = 161
value = [0, 0, 0, 128, 33, 0]
class = Präsens
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1]
class = other
lemmata1=denken? <= 0.5
gini = 0.489
samples = 135
value = [0, 0, 0, 80, 54, 1]
class = Präsens
gini = 0.0
samples = 3
value = [0, 0, 0, 0, 3, 0]
class = Präteritum
lemmata2=rob? <= 0.5
gini = 0.056
samples = 526
value = [0, 511, 11, 0, 0, 4]
class = Perfekt
gini = 0.0
samples = 1
value = [0, 0, 0, 1, 0, 0]
class = Präsens
lemmata1=kriegen? <= 0.5
gini = 0.32
samples = 160
value = [0, 0, 0, 128, 32, 0]
class = Präsens
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 1, 0]
class = Präteritum
lemmata1=folgen? <= 0.5
gini = 0.483
samples = 130
value = [0, 0, 0, 79, 50, 1]
class = Präsens
gini = 0.32
samples = 5
value = [0, 0, 0, 1, 4, 0]
class = Präteritum
lemmata2=annufen? <= 0.5
gini = 0.052
samples = 525
value = [0, 511, 11, 0, 0, 3]
class = Perfekt
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1]
class = other
pos1=VAFIN? <= 0.5
gini = 0.314
samples = 159
value = [0, 0, 0, 128, 31, 0]
class = Präsens
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 1, 0]
class = Präteritum
lemmata1=finden? <= 0.5
gini = 0.475
samples = 125
value = [0, 0, 0, 78, 46, 1]
class = Präsens
gini = 0.32
samples = 5
value = [0, 0, 0, 1, 4, 0]
class = Präteritum
gini = 0.049
samples = 524
value = [0, 511, 11, 0, 0, 2]
class = Perfekt
gini = 0.0
samples = 1
value = [0, 0, 0, 0, 0, 1]
class = other
gini = 0.0
samples = 21
value = [0, 0, 0, 21, 0, 0]
class = Präsens
gini = 0.348
samples = 138
value = [0, 0, 0, 107, 31, 0]
class = Präsens
gini = 0.47
samples = 123
value = [0, 0, 0, 78, 44, 1]
class = Präsens
gini = 0.0
samples = 2
value = [0, 0, 0, 0, 0, 2, 0]
class = Präteritum