Busca e classificação ordenada (ranking) TEC

Recuperação e Ranqueamento de Texto, Análise de Dados e Aprendizagem Supervisionada com base no texto da Tarifa Externa Comum do Mercosul

Resumo

Utilização das técnicas word2vec, BagofWords, unigrama, bigrama, totemização, text similarity, OkapiBM25+, entre outras, para processamento dos subitens passíveis de utilização para sugestão de classificação fiscal conforme descrito no texto da TEC Mercosul e ranqueamento com relação a uma lista de palavras buscadas. Comparação com métodos de aprendizado de máquina.

Introdução

A Classificação Fiscal correta é sempre um desafio. As edições brasileiras do texto da Tarifa Externa Comum tem dezenas de milhares de palavras, uma centena de capítulos, mais de 10.000 subitens passíveis de uso para classificação, uma hierarquia complexa e nem sempre uniforme, além de Notas de Seção, Notas de Capítulo, Regras gerais, etc. Neste trabalho é proposto o uso computadores para levantar estes números exatos para nós, pois nisso os computadores já superam os seres humanos (e com os últimos avanços em IA e Aprendizado de máquina -AI Machine Learning e Deep Learning estão começando a nos alcançar no processamento estatístico de textos e na visão e reconhecimento de objetos, em alguns casos superando).

Este artigo está estruturado em várias partes, relativamente independentes.

Análise do texto da TEC e considerações sobre o desafio da Classificação Fiscal. Primeiros processamentos de texto e escolha de um caminho.

Testes e estatísticas na linguagem de programação Python(com código). Aplicação das técnicas de vetorização, bigrama, stemização, Query similarity e OkapiBM25+.

Testes de modelos de aprendizado estatístico para tarefa similar.

Importar bibliotecas necessárias

In [2]:

```
import numpy as np
import nltk
import sklearn
import sys

sys.path.insert(0, '..')

nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /home/ivan/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[2]:

```
True
```

Parte 1 - Ler TEC e montar documentos

Abrindo a TEC no Word, podemos começar a analisar sua estrutura. Basicamente, ela possui seis partes:

- Títulos de Seções e Capítulos
- Abreviaturas e Símbolos
- Regras Gerais para Interpretação do Sistema Harmonizado
- Regras Gerais Complementares
- Regra de Tributação para Produtos do Setor Aeronáutico
- Nomenclatura Comum do MERCOSUL (NCM) e Regime Tarifário Comum

Para nosso trabalho, o importante é a sexta parte, que por sua vez assim se divide:

- Seção (Descrição)
- Notas de Seção
- Capítulo (Descrição)
- Notas de Capítulo
- Tabela NCM, Descrição, TEC

Cabe aqui um adendo e uma observação importante: a tarefa de classificação fiscal é um trabalho técnico, que pode depender de outras informações e conhecimentos específicos, que não estarão escritos no texto da TEC, como por exemplo conhecimentos sobre metalurgia, química ou têxteis. Além também de conhecimento sobre as regras de classificação fiscal (descritas na TEC), interpretação gramatical e legal, decomposição do produto caso misto, entre outros. Por isso a legislação determinou competência à autoridade tributária (O Auditor Fiscal em exercício na Receita Federal) em termos de decisão em classificação fiscal de mercadorias (artigo 15, inciso XIX, do Decreto 7.482/11 e INSTRUÇÃO NORMATIVA RFB Nº 1464, DE 08 DE MAIO DE 2014, entre outros).

Cabe lembrar também que o que ora chamamos de “TEC” é uma compilação de tratados, tradições, estudos técnicos e padrões internacionais compilados primeiramente no sistema harmonizado internacional (SH) na chamada Convenção Internacional de Bruxelas, com chancela da OMA (Organização Mundial das Aduanas) e depois expandidos para a experiência e necessidades do Mercosul NCM — Mercado Comum do Mercosul Nomenclatura Comum de Mercadorias. Além disso, no Brasil, vigoram também regras federais adicionais do RIPI — Regulamento do Imposto sobre Produtos Industrializados bem como regulamentos do ICMS — Estaduais.

A solução, no âmbito deste trabalho, é fazer uma “pontuação” dos subitens da TEC e exibir os resultados em ordem decrescente. Afinal, são mais de 10.000 possibilidades. Então faremos o computador passar o que podemos chamar de uma “peneira grossa” e analisaremos apenas umas poucas posições melhor ranqueadas. Talvez precisemos analisar apenas as três ou quatro primeiras da lista se tivermos “sorte”.

Por “sorte”, podemos definir a pergunta “certa” também. No caso, a pergunta seria a descrição da mercadoria que se quer classificar. Assim, a autoridade (Auditor Fiscal), o Importador, o Fabricante, o Assistente Técnico, Despachante Aduaneiro, preposto, ou outro que possua a responsabilidade de descrever a mercadoria deve fazê-lo de modo a permitir identificar corretamente o produto e definir sua correta classificação fiscal (IN RFB 680/2006 e Regulamento Aduaneiro Art. 711, entre outros).

Em resumo, podemos didaticamente então “fatiar” a tarefa classificação fiscal em duas etapas: 1. Descrever o produto; 2. Encaixar a descrição na classificação correta. Como a TEC/NCM é hierárquica, podemos fazer uma analogia com uma árvore. Normalmente, percorremos os ramos desta árvore em busca do posicionamento correto. Mas há possibilidade de estarmos olhando para um ramo e o item correto estar em outro, e não o enxergarmos por estarmos no ramo errado usando um subitem muito parecido. Superar esta busca hierárquica é o objetivo deste sistema.

Fazendo outra analogia, pensemos em um egiptólogo: no passo 1 ele define exatamente o artefato que a expedição irá buscar, e descreve, através de pesquisas anteriores, o caminho: Como é o artefato, em que sítio arqueológico estaria, etc. O segundo passo é a expedição propriamente dita. Nesta, a equipe se enreda por galerias, túneis, etc. E pode achar um artefato MUITO parecido em uma câmara, mas não olhar na câmara vizinha, onde está o artefato realmente procurado. Para evitar esta situação, nosso robô percorrerá todas as câmaras e pontuará os artefatos encontrados de acordo com a similaridade com o artefato descrito no primeiro passo. Esta pontuação será baseada em uma relação de similaridade entre uma frase que descreve o que se busca e cada um dos subitens da TEC.

No caso da classificação fiscal há um terceiro passo ainda, pois uma posição ou capítulo ou outro pode ser excluído ou pode haver uma condicional imposta por uma nota de seção ou capítulo ou outro tipo de exceção, como uma das regras gerais. Esta tarefa poderia ser passível de automatização também através de técnicas de NLP, mas ainda não é o escopo deste trabalho.

Agora, o que podemos entender por “subitens da TEC”. Analisando a parte “Tabela NCM, Descrição, TEC”, vemos que nem todas as linhas possuem valor na coluna “TEC”:

Primeira página da TEC, versão Word. Note-se a terceira coluna da tabela

In [4]:

```
from IPython.display import Image
Image("img/tec.png")
```

Out[4]:

Capítulo 1

Animais vivos

Nota.

1.- O presente Capítulo compreende todos os animais vivos, exceto:

a) Peixes e crustáceos, moluscos e outros invertebrados aquáticos, das posições 03.01, 03.06, 03.07 ou 03.08;

b) Culturas de microrganismos e os outros produtos da posição 30.02;

c) Animais da posição 95.08.

NCM	DESCRIÇÃO	TEC (%)
01.01	**Cavalos, asininos e muares, vivos.**	
0101.2	- Cavalos:	
0101.21.00	-- Reprodutores de raça pura	0
0101.29.00	-- Outros	2
0101.30.00	- Asininos	4
0101.90.00	- Outros	4
01.02	**Animais vivos da espécie bovina.**	
0102.2	- Bovinos domésticos:	
0102.21	-- Reprodutores de raça pura	
0102.21.10	Prenhes ou com cria ao pé	0
0102.21.90	Outros	0
0102.29	-- Outros	
0102.29.1	Para reprodução	
0102.29.11	Prenhes ou com cria ao pé	2

Na tela, notamos facilmente a estrutura antes descrita (Capítulo XX (Descrição), Notas de Capítulo, Tabela NCM. Também notamos que apenas algumas linhas da tabela possuem valores na terceira coluna (TEC %). Ocorre que a estrutura da NCM é hierarquizada, na sequência do SH: Capítulo(2 dígitos), Posição(4 dígitos), Subposição 1 e 2(5 e 6 dígitos). O Mercosul adiciona ainda mais dois dígitos: Item e Subitem (7º e 8º dígitos respectivamente). Apenas a linha completa, de maior ordem, contendo uma alíquota TEC (assim como nos textos próprios alíquotas de IPI e ICMS) pode ser usada para classificação fiscal.

Mas, olhando novamente na tabela, vemos que o campo descrição nestas linhas normalmente é espartano. Por exemplo, o subitem NCM 0102.21.10 fala apenas em “prenhes ou com cria ao pé” e, pior, o seguinte, 0102.21.90, apenas “Outros”.

Assim, vamos criar um conceito chamado “Lista TEC com descrição completa”. Desta forma, cada subposição, item ou subitem deve ser descrito com a concatenação da sua árvore hieráquica. Assim, sabemos que o subitem 0102.21.10 refere-se a “Animais vivos da espécie bovina — Bovinos domésticos — Reprodutores de raça pura Prenhes com cria ao pé” e o subitem 0102.31.90 refere-se a “Animais vivos da espécie bovina — Búfalos — Reprodutores de raça pura Outros”. Certamente bem mais informativo que antes. Podemos ainda pensar em somar as descrições dos Capítulos nesta hierarquia.

Embora cada linha fique mais informativa, a quantidade total de palavras a ser buscada ficará muito maior, pois muitas se repetirão. Além disso, a quantidade de palavras em algumas descrições da tabela já é grande por si só, e concatenar com mais descrições tornará tudo maior. Mas não podemos perder o foco: o objetivo é produzir linhas para que o computador, e não seres humanos, procurem e contem ocorrências de palavras. E nisso os computadores são imensamente melhores que nós, seres humanos. Esta linha que está sendo proposta será “digerida” e (pré)processada por máquinas digitais, que retornarão ao ser humano apenas o “filé” que este busca dentre as mais de 10.000 opções.

Foram criadas algumas funções/scripts para fazer este pré-processamento, conforme abaixo:

In [7]:

```
import batch.processtec as pt

listaTEC = pt.leTEC()
print('Texto original do arquivo:')
print(listaTEC[:60])
```

```
Texto original do arquivo:
['\n', '\n', '\n', '\n', '\n', '\n', '\n', '\n', '\n', '\n', '\n',
'\n', '\n', '\n', '\n', 'NOMENCLATURA COMUM DO MERCOSUL (NCM)\n', 'E
TARIFA EXTERNA COMUM (TEC)\n', 'BRASIL \n', '2 0 1 7\n', 'Atualizada
até a Resolução Camex nº 15 de 17/02/2017 (DOU 21/02/2017)\n', '\n',
'(Baseada no Sistema Harmonizado de Designação e de \n', 'Codificaçã
o de Mercadorias, atualizado com sua VI Emenda)\n', '\n', '\n',
'\n', '\n', '\n', '\n', '\n', '\n', '\n', '\n', '\n', '\n', 'CONTEÚD
O\n', '\n', '\n', '\n', '\n', 'Títulos de Seções e Capítulos\n', 'Ab
reviaturas e Símbolos\n', 'Regras Gerais para Interpretação do Siste
ma Harmonizado\n', 'Regras Gerais Complementares\n', 'Regra de Tribu
tação para Produtos do Setor Aeronáutico\n', 'Nomenclatura Comum do
MERCOSUL (NCM) e Regime Tarifário Comum\n', '\n', '\n', '\n', '\n',
'\n', '\n', '\n', '\n', '\n', 'Notas.\n', '\n', 'Na Nomenclatura, os
termos e expressões seguidos de um asterisco e que constam entre par
ênteses, são equivalentes aos que precedem, e são utilizados nos dem
ais países de língua portuguesa.\n', '\n', 'BK\tNa Nomenclatura, est
a sigla identifica as mercadorias definidas como Bens de Capita
l.\n']
```

In [8]:

```
listaNCM = pt.montaNCM(listaTEC)
print('Linhas originais da tabela da TEC:')
for itemTEC in listaNCM[:10]:
    print(itemTEC)
```

```
Linhas originais da tabela da TEC:
['01.01', 'Cavalos, asininos e muares, vivos.', '']
['0101.2', '-\tCavalos:', '']
['0101.21.00', '--\tReprodutores de raça pura', '0']
['0101.29.00', '--\tOutros', '2']
['0101.30.00', '-\tAsininos', '4']
['0101.90.00', '-\tOutros', '4']
['01.02', 'Animais vivos da espécie bovina.', '']
['0102.2', '-\tBovinos domésticos:', '']
['0102.21', '--\tReprodutores de raça pura', '']
['0102.21.10', 'Prenhes ou com cria ao pé', '0']
```

In [10]:

```
listaTECResumo = pt.montaTECResumo(listaNCM)
print('TEC Resumo - linhas dos subitens com descrição dos nós pais incluída:')
for itemTEC_descricao_completa in listaTECResumo[:10]:
    print(itemTEC_descricao_completa)
```

```
TEC Resumo - linhas dos subitens com descrição dos nós pais incluíd
a:
0101.21.00 --   Reprodutores de raça pura -     Cavalos: Cavalos, as
ininos e muares, vivos.
0101.29.00 --   Outros -        Cavalos: Cavalos, asininos e muares,
vivos.
0101.30.00 -    Asininos Cavalos, asininos e muares, vivos.
0101.90.00 -    Outros Cavalos, asininos e muares, vivos.
0102.21.10 Prenhes ou com cria ao pé -- Reprodutores de raça pura -
Bovinos domésticos: Animais vivos da espécie bovina.
0102.21.90 Outros --    Reprodutores de raça pura -     Bovinos domé
sticos: Animais vivos da espécie bovina.
0102.29.11 Prenhes ou com cria ao pé Para reprodução -- Outros -
Bovinos domésticos: Animais vivos da espécie bovina.
0102.29.19 Outros Para reprodução --    Outros -        Bovinos domé
sticos: Animais vivos da espécie bovina.
0102.29.90 Outros --    Outros -        Bovinos domésticos: Animais
vivos da espécie bovina.
0102.31.10 Prenhes ou com cria ao pé -- Reprodutores de raça pura -
Búfalos: Animais vivos da espécie bovina.
```

In [12]:

```
corpus = []

for itemTEC_descricao_completa in listaTECResumo:
    documento = pt.tokenize_to_words(itemTEC_descricao_completa)
    corpus.append(' '.join(documento[1:]))
```

In [14]:

```
print('Corpus - descrição completa sem stopwords, sem caracteres especias e pont
uação, somente minúsculas')
corpus[:100]
```

Corpus - descrição completa sem stopwords, sem caracteres especias e pontuação, somente minúsculas

Out[14]:

```
['reprodutores raca pura cavalos cavalos asininos muares vivos',
 'outros cavalos cavalos asininos muares vivos',
 'asininos cavalos asininos muares vivos',
 'outros cavalos asininos muares vivos',
 'prenhes cria reprodutores raca pura bovinos domesticos animais viv
os especie bovina',
 'outros reprodutores raca pura bovinos domesticos animais vivos esp
ecie bovina',
 'prenhes cria para reproducao outros bovinos domesticos animais viv
os especie bovina',
 'outros para reproducao outros bovinos domesticos animais vivos esp
ecie bovina',
 'outros outros bovinos domesticos animais vivos especie bovina',
 'prenhes cria reprodutores raca pura bufalos animais vivos especie
bovina',
 'outros reprodutores raca pura bufalos animais vivos especie bovin
a',
 'prenhes cria para reproducao outros bufalos animais vivos especie
bovina',
 'outros para reproducao outros bufalos animais vivos especie bovin
a',
 'outros outros bufalos animais vivos especie bovina',
 'outros animais vivos especie bovina',
 'reprodutores raca pura animais vivos especie suina',
 'de peso inferior 50 kg outros animais vivos especie suina',
 'de peso igual superior 50 kg outros animais vivos especie suina',
 'prenhes cria reprodutores raca pura ovinos animais vivos especies
ovina caprina',
 'outros reprodutores raca pura ovinos animais vivos especies ovina
caprina',
 'outros ovinos animais vivos especies ovina caprina',
 'reprodutores raca pura caprinos animais vivos especies ovina capri
na',
 'outros caprinos animais vivos especies ovina caprina',
 'linhas puras hibridas reproducao aves especie gallus domesticus de
peso superior aves especie gallus domesticus patos gansos perus peru
as galinhas d angola pintadas especies domesticas vivos',
 'outros aves especie gallus domesticus de peso superior aves especi
e gallus domesticus patos gansos perus peruas galinhas d angola pint
adas especies domesticas vivos',
 'peruas perus de peso superior aves especie gallus domesticus patos
gansos perus peruas galinhas d angola pintadas especies domesticas v
ivos',
 'patos de peso superior aves especie gallus domesticus patos gansos
perus peruas galinhas d angola pintadas especies domesticas vivos',
 'gansos de peso superior aves especie gallus domesticus patos ganso
s perus peruas galinhas d angola pintadas especies domesticas vivo
s',
 'galinhas d angola pintadas de peso superior aves especie gallus do
mesticus patos gansos perus peruas galinhas d angola pintadas especi
es domesticas vivos',
 'aves especie gallus domesticus outros aves especie gallus domestic
us patos gansos perus peruas galinhas d angola pintadas especies dom
esticas vivos',
 'outros outros aves especie gallus domesticus patos gansos perus pe
ruas galinhas d angola pintadas especies domesticas vivos',
 'primatas mamiferos outros animais vivos',
 'baleias golfinhos botos mamiferos ordem cetacea peixes boi manatin
s dugongos mamiferos ordem sirenia otarias focas leoes marinhos mors
```

```
as mamiferos subordem pinnipedia mamiferos outros animais vivos',
 'camelos outros camelideos camelidae mamiferos outros animais vivo
s',
 'coelhos lebres mamiferos outros animais vivos',
 'outros mamiferos outros animais vivos',
 'repteis incluindo serpentes tartarugas marinhas outros animais viv
os',
 'aves rapina aves outros animais vivos',
 'psitaciformes incluindo papagaios periquitos araras catatuas aves
outros animais vivos',
 'avestruzes struthio camelus reproducao avestruzes emus dromaius no
vaehollandiae aves outros animais vivos',
 'outros avestruzes emus dromaius novaehollandiae aves outros animai
s vivos',
 'outras aves outros animais vivos',
 'abelhas insetos outros animais vivos',
 'outros insetos outros animais vivos',
 'outros outros animais vivos',
 'carcacas meias carcacas carnes animais especie bovina frescas refr
igeradas',
 'quartos dianteiros outras pecas desossadas carnes animais especie
bovina frescas refrigeradas',
 'quartos traseiros outras pecas desossadas carnes animais especie b
ovina frescas refrigeradas',
 'outras outras pecas desossadas carnes animais especie bovina fresc
as refrigeradas',
 'desossadas carnes animais especie bovina frescas refrigeradas',
 'carcacas meias carcacas carnes animais especie bovina congeladas',
 'quartos dianteiros outras pecas desossadas carnes animais especie
bovina congeladas',
 'quartos traseiros outras pecas desossadas carnes animais especie b
ovina congeladas',
 'outras outras pecas desossadas carnes animais especie bovina conge
ladas',
 'desossadas carnes animais especie bovina congeladas',
 'carcacas meias carcacas frescas refrigeradas carnes animais especi
e suina frescas refrigeradas congeladas',
 'pernas respectivos pedacos desossados frescas refrigeradas carnes
animais especie suina frescas refrigeradas congeladas',
 'outras frescas refrigeradas carnes animais especie suina frescas r
efrigeradas congeladas',
 'carcacas meias carcacas congeladas carnes animais especie suina fr
escas refrigeradas congeladas',
 'pernas respectivos pedacos desossados congeladas carnes animais es
pecie suina frescas refrigeradas congeladas',
 'outras congeladas carnes animais especie suina frescas refrigerada
s congeladas',
 'carcacas meias carcacas cordeiro frescas refrigeradas carnes anima
is especies ovina caprina frescas refrigeradas congeladas',
 'carcacas meias carcacas outras carnes animais especie ovina fresca
s refrigeradas carnes animais especies ovina caprina frescas refrige
radas congeladas',
 'outras pecas desossadas outras carnes animais especie ovina fresca
s refrigeradas carnes animais especies ovina caprina frescas refrige
radas congeladas',
 'desossadas outras carnes animais especie ovina frescas refrigerada
s carnes animais especies ovina caprina frescas refrigeradas congela
das',
 'carcacas meias carcacas cordeiro congeladas carnes animais especie
s ovina caprina frescas refrigeradas congeladas',
 'carcacas meias carcacas outras carnes animais especie ovina congel
```

```
adas carnes animais especies ovina caprina frescas refrigeradas cong
eladas',
 'outras pecas desossadas outras carnes animais especie ovina congel
adas carnes animais especies ovina caprina frescas refrigeradas cong
eladas',
 'desossadas outras carnes animais especie ovina congeladas carnes a
nimais especies ovina caprina frescas refrigeradas congeladas',
 'carnes animais especie caprina carnes animais especies ovina capri
na frescas refrigeradas congeladas',
 'carnes animais especies cavalar asinina muar frescas refrigeradas
congeladas',
 'da especie bovina frescas refrigeradas miudezas comestiveis animai
s especies bovina suina ovina caprina cavalar asinina muar frescas r
efrigeradas congeladas',
 'linguas da especie bovina congeladas miudezas comestiveis animais
especies bovina suina ovina caprina cavalar asinina muar frescas ref
rigeradas congeladas',
 'figados da especie bovina congeladas miudezas comestiveis animais
especies bovina suina ovina caprina cavalar asinina muar frescas ref
rigeradas congeladas',
 'rabos outras da especie bovina congeladas miudezas comestiveis ani
mais especies bovina suina ovina caprina cavalar asinina muar fresca
s refrigeradas congeladas',
 'outros outras da especie bovina congeladas miudezas comestiveis an
imais especies bovina suina ovina caprina cavalar asinina muar fresc
as refrigeradas congeladas',
 'da especie suina frescas refrigeradas miudezas comestiveis animais
especies bovina suina ovina caprina cavalar asinina muar frescas ref
rigeradas congeladas',
 'figados da especie suina congeladas miudezas comestiveis animais e
species bovina suina ovina caprina cavalar asinina muar frescas refr
igeradas congeladas',
 'outras da especie suina congeladas miudezas comestiveis animais es
pecies bovina suina ovina caprina cavalar asinina muar frescas refri
geradas congeladas',
 'outras frescas refrigeradas miudezas comestiveis animais especies
bovina suina ovina caprina cavalar asinina muar frescas refrigeradas
congeladas',
 'outras congeladas miudezas comestiveis animais especies bovina sui
na ovina caprina cavalar asinina muar frescas refrigeradas congelada
s',
 'nao cortadas pedacos frescas refrigeradas de aves especie gallus d
omesticus carnes miudezas comestiveis frescas refrigeradas congelada
s aves posicao 01 05',
 'nao cortadas pedacos congeladas de aves especie gallus domesticus
carnes miudezas comestiveis frescas refrigeradas congeladas aves pos
icao 01 05',
 'pedacos miudezas frescos refrigerados de aves especie gallus domes
ticus carnes miudezas comestiveis frescas refrigeradas congeladas av
es posicao 01 05',
 'pedacos miudezas congelados de aves especie gallus domesticus carn
es miudezas comestiveis frescas refrigeradas congeladas aves posicao
01 05',
 'nao cortadas pedacos frescas refrigeradas de peruas perus carnes m
iudezas comestiveis frescas refrigeradas congeladas aves posicao 01
05',
 'nao cortadas pedacos congeladas de peruas perus carnes miudezas co
mestiveis frescas refrigeradas congeladas aves posicao 01 05',
 'pedacos miudezas frescos refrigerados de peruas perus carnes miude
zas comestiveis frescas refrigeradas congeladas aves posicao 01 05',
 'pedacos miudezas congelados de peruas perus carnes miudezas comest
```

```
iveis frescas refrigeradas congeladas aves posicao 01 05',
 'nao cortadas pedacos frescas refrigeradas de patos carnes miudezas
comestiveis frescas refrigeradas congeladas aves posicao 01 05',
 'nao cortadas pedacos congeladas de patos carnes miudezas comestive
is frescas refrigeradas congeladas aves posicao 01 05',
 'figados gordos foies gras frescos refrigerados de patos carnes miu
dezas comestiveis frescas refrigeradas congeladas aves posicao 01 0
5',
 'outras frescas refrigeradas de patos carnes miudezas comestiveis f
rescas refrigeradas congeladas aves posicao 01 05',
 'outras congeladas de patos carnes miudezas comestiveis frescas ref
rigeradas congeladas aves posicao 01 05',
 'nao cortadas pedacos frescas refrigeradas de gansos carnes miudeza
s comestiveis frescas refrigeradas congeladas aves posicao 01 05',
 'nao cortadas pedacos congeladas de gansos carnes miudezas comestiv
eis frescas refrigeradas congeladas aves posicao 01 05',
 'figados gordos foies gras frescos refrigerados de gansos carnes mi
udezas comestiveis frescas refrigeradas congeladas aves posicao 01 0
5',
 'outras frescas refrigeradas de gansos carnes miudezas comestiveis
frescas refrigeradas congeladas aves posicao 01 05',
 'outras congeladas de gansos carnes miudezas comestiveis frescas re
frigeradas congeladas aves posicao 01 05',
 'de galinhas d angola pintadas carnes miudezas comestiveis frescas
refrigeradas congeladas aves posicao 01 05']
```

In [15]:

```
listaCapitulos = pt.montaCapitulos(listaTEC)
for capitulo in listaCapitulos[:3]:
    print(capitulo)
```

```
['Capítulo 1', 'Animais vivos', '1.-\tO presente Capítulo compreende
todos os animais vivos, exceto:\n\na)\tPeixes e crustáceos, moluscos
e outros invertebrados aquáticos, das posições 03.01, 03.06, 03.07 o
u 03.08;\n\nb)\tCulturas de microrganismos e os outros produtos da p
osição 30.02;\n\nc)\tAnimais da posição 95.08.\n\n']
['Capítulo 2', 'Carnes e miudezas, comestíveis', '1.-\tO presente Ca
pítulo não compreende:\n\na)\tNo que diz respeito às posições 02.01
a 02.08 e 02.10, os produtos impróprios para alimentação humana;\n\n
b)\tAs tripas, bexigas e estômagos, de animais (posição 05.04), nem
o sangue animal (posições 05.11 ou 30.02);\n\nc)\tAs gorduras animai
s, exceto os produtos da posição 02.09 (Capítulo 15).\n\n']
['Capítulo 3', 'Peixes e crustáceos, moluscos e \n outros invertebra
dos aquáticos', '1.-\tO presente Capítulo não compreende:\n\na)\tOs
mamíferos da posição 01.06;\n\nb)\tAs carnes dos mamíferos da posiçã
o 01.06 (posições 02.08 ou 02.10);\n\nc)\tOs peixes (incluindo os se
us fígados, ovas e gônadas masculinas) e crustáceos, moluscos e outr
os invertebrados aquáticos, mortos e impróprios para alimentação hum
ana, seja pela sua natureza, seja pelo seu estado de apresentação (C
apítulo 5); as farinhas, pós e pellets de peixes ou de crustáceos, d
e moluscos ou de outros invertebrados aquáticos, impróprios para ali
mentação humana (posição 23.01);\n\nd)\tO caviar e seus sucedâneos p
reparados a partir de ovas de peixe (posição 16.04).\n\n2.-\tNo pres
ente Capítulo, o termo \x93pellets\x94 designa os produtos apresenta
dos sob a forma de cilindros, bolas, etc., aglomerados quer por simp
les pressão, quer pela adição de um aglutinante em pequena quantidad
e.\n\n']
```

Usar sklearn para montar o DTM

Utilizando scikit-learn, montar Bag of Words do Corpus criado

In [18]:

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(analyzer='word')

X = vectorizer.fit_transform(corpus)
```

In [19]:

```
len(vectorizer.get_feature_names())
```

Out[19]:

9369

In [20]:

```
len(corpus)
```

Out[20]:

10147

In [23]:

```
print('O corpus montado a partir de TEC resumo tem '
      '%d documentos e %d palavras diferentes' %
        X.toarray().shape)
```

```
O corpus montado a partir de TEC resumo tem 10147 documentos e 9369
palavras diferentes
```

In [30]:

```
print('Visualizar parte do vocabulário: ')
vectorizer.get_feature_names()[200:300]
```

Visualizar parte do vocabulário:

Out[30]:

```
['8443',
 '8447',
 '8450',
 '8451',
 '8456',
 '8462',
 '8470',
 '8471',
 '8472',
 '8473',
 '8481',
 '85',
 '8501',
 '8504',
 '8523',
 '8525',
 '8526',
 '8543',
 '86',
 '8606',
 '87',
 '8701',
 '8704',
 '88',
 '89',
 '8901',
 '90',
 '900',
 '9010',
 '9011',
 '9015',
 '9030',
 '91',
 '92',
 '93',
 '94',
 '940',
 '95',
 '9503',
 '9504',
 '96',
 '960',
 '9603',
 '9608',
 '97',
 '98',
 '99',
 'aba',
 'abaca',
 'abacates',
 'abacavir',
 'abacaxi',
 'abacaxis',
 'abajures',
 'abalones',
 'abelha',
 'abelhas',
 'abertas',
 'abertos',
```

```
 'abertura',
 'abeto',
 'abies',
 'abietatos',
 'aboboras',
 'abobrinhas',
 'abotoaduras',
 'abrangido',
 'abrasivos',
 'abre',
 'abridoras',
 'abrigos',
 'abrir',
 'abrotea',
 'abroteas',
 'abrunhos',
 'abs',
 'absolutos',
 'absorcao',
 'absorvente',
 'absorventes',
 'absorver',
 'absorvido',
 'absorviveis',
 'acabadas',
 'acabados',
 'acabamento',
 'acabar',
 'acafrao',
 'acai',
 'acampamento',
 'acampar',
 'acamurcados',
 'acanthistius',
 'acao',
 'acaricidas',
 'acefato',
 'acelerador',
 'aceleradores',
 'acendedores',
 'acer']
```

In [25]:

```
X[:10].toarray()
```

Out[25]:

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

Fazer processamento TFIDF

In [32]:

```
from sklearn.feature_extraction.text import TfidfTransformer

tfid = TfidfTransformer()

Xtfid = tfid.fit_transform(X)
```

In [33]:

```
Xtfid
```

Out[33]:

```
<10147x9369 sparse matrix of type '<class 'numpy.float64'>'
        with 180484 stored elements in Compressed Sparse Row format>
```

In [34]:

```
Xtfid[:10].toarray()
```

Out[34]:

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

In [35]:

```
for ind, (word, index) in enumerate(vectorizer.vocabulary_.items()):
    print(word, index)
    if ind > 10:
        break
```

```
reprodutores 7761
raca 7553
pura 7491
cavalos 1896
asininos 969
muares 6230
vivos 9284
outros 6595
prenhes 7321
cria 2655
bovinos 1413
domesticos 3281
```

Exibir nuvens de palavras e visualizar palavras frequentes

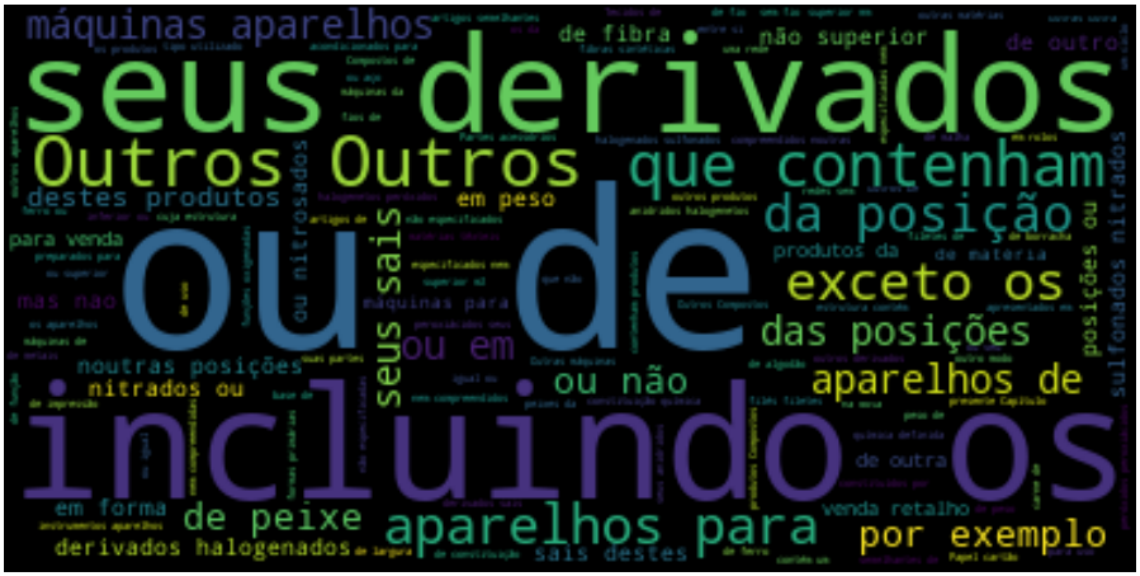
In [36]:

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

In [37]:

```
# Create and generate a word cloud image:
wordcloud = WordCloud().generate(' '.join(listaTECResumo))

# Display the generated image:
plt.figure(figsize=(16, 12))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



In [38]:

```
stopwords = set(STOPWORDS)
stopwords.update(['os', 'de', 'seus', 'que', "das", "derivados", 'não', 'posiçã
o',
                  'Outros', 'outro', 'Outro', 'ou'])

# Generate a word cloud image
wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate('
 '.join(listaTECResumo))
plt.figure(figsize=(16, 12))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Com o sklearn, fazer uma "busca" por uma palavra e confirmar sua pontuação em X e em Xtfidf

In [21]:

```
def get_index_documentos_palavra(X, palavra: str):
    indpalavra = vectorizer.vocabulary_[palavra]
    return np.argwhere(X[:, indpalavra].toarray() > 0), indpalavra

def show_dados_palavra(documentos, X, Xtfidf, palavra:str):
    indexes, indpalavra = get_index_documentos_palavra(X, palavra)
    for linha in indexes:
        inddocumento = linha[0]
        print(documentos[inddocumento])
        print(Xtfid[inddocumento, indpalavra])
        print(X[inddocumento, indpalavra])
```

In [22]:

```
indexes, indpalavra = get_index_documentos_palavra(X, 'arruelas')
print(indexes, indpalavra)
```

```
[[4756    0]
 [4803    0]
 [4804    0]
 [7097    0]
 [7098    0]
 [7099    0]
 [7100    0]
 [7101    0]
 [7102    0]
 [7103    0]
 [7104    0]
 [7105    0]
 [7106    0]
 [7107    0]
 [7108    0]
 [7205    0]
 [7206    0]
 [7207    0]
 [7208    0]
 [7209    0]
 [7293    0]] 936
```

In [23]:

```
show_dados_palavra(corpus, X, Xtfid, 'arruelas')
```

arruelas outras outras obras plastico obras outras materias posicoes
39 01 39 14
0.35209044782641996
1
perfis recauchutagem outras formas por exemplo varetas tubos perfis
artigos por exemplo discos arruelas anilhas borracha vulcanizada
0.2874789688267496
1
outros outras formas por exemplo varetas tubos perfis artigos por ex
emplo discos arruelas anilhas borracha vulcanizada
0.3380702594906789
1
tira fundos artigos roscados parafusos pinos pernos roscados porcas
tira fundos ganchos roscados rebites chavetas cavilhas contrapinos t
rocos arruelas anilhas incluindo pressao artigos semelhantes ferro f
undido ferro aco
0.17171849214498303
1
outros parafusos madeira artigos roscados parafusos pinos pernos ros
cados porcas tira fundos ganchos roscados rebites chavetas cavilhas
contrapinos trocos arruelas anilhas incluindo pressao artigos semelh
antes ferro fundido ferro aco
0.18128620627348382
1
ganchos armelas pitoes artigos roscados parafusos pinos pernos rosca
dos porcas tira fundos ganchos roscados rebites chavetas cavilhas co
ntrapinos trocos arruelas anilhas incluindo pressao artigos semelhan
tes ferro fundido ferro aco
0.1720059105961564
1
parafusos perfurantes artigos roscados parafusos pinos pernos roscad
os porcas tira fundos ganchos roscados rebites chavetas cavilhas con
trapinos trocos arruelas anilhas incluindo pressao artigos semelhant
es ferro fundido ferro aco
0.1776287014679304
1
outros parafusos pinos pernos porcas arruelas anilhas artigos roscad
os parafusos pinos pernos roscados porcas tira fundos ganchos roscad
os rebites chavetas cavilhas contrapinos trocos arruelas anilhas inc
luindo pressao artigos semelhantes ferro fundido ferro aco
0.2996100476868767
2
porcas artigos roscados parafusos pinos pernos roscados porcas tira
fundos ganchos roscados rebites chavetas cavilhas contrapinos trocos
arruelas anilhas incluindo pressao artigos semelhantes ferro fundido
ferro aco
0.1827727597519414
1
outros artigos roscados parafusos pinos pernos roscados porcas tira
fundos ganchos roscados rebites chavetas cavilhas contrapinos trocos
arruelas anilhas incluindo pressao artigos semelhantes ferro fundido
ferro aco
0.1916006548870102
1
arruelas anilhas pressao outras arruelas anilhas seguranca artigos r
oscados parafusos pinos pernos roscados porcas tira fundos ganchos r
oscados rebites chavetas cavilhas contrapinos trocos arruelas anilha
s incluindo pressao artigos semelhantes ferro fundido ferro aco
0.44070400092879897
3
outras arruelas anilhas artigos roscados parafusos pinos pernos rosc

ados porcas tira fundos ganchos roscados rebites chavetas cavilhas c
ontrapinos trocos arruelas anilhas incluindo pressao artigos semelha
ntes ferro fundido ferro aco
0.3463490721909246
2
rebites artigos roscados parafusos pinos pernos roscados porcas tira
fundos ganchos roscados rebites chavetas cavilhas contrapinos trocos
arruelas anilhas incluindo pressao artigos semelhantes ferro fundido
ferro aco
0.1826026978901023
1
chavetas cavilhas contrapinos trocos artigos roscados parafusos pino
s pernos roscados porcas tira fundos ganchos roscados rebites chavet
as cavilhas contrapinos trocos arruelas anilhas incluindo pressao ar
tigos semelhantes ferro fundido ferro aco
0.15873355931187477
1
outros artigos roscados parafusos pinos pernos roscados porcas tira
fundos ganchos roscados rebites chavetas cavilhas contrapinos trocos
arruelas anilhas incluindo pressao artigos semelhantes ferro fundido
ferro aco
0.1916006548870102
1
tachas pregos percevejos escapulas artigos semelhantes tachas pregos
percevejos escapulas artigos semelhantes cobre ferro cabeca cobre pa
rafusos pinos pernos roscados porcas ganchos roscados rebites chavet
as cavilhas contrapinos trocos arruelas anilhas incluindo pressao ar
tigos semelhantes cobre
0.14737616600525336
1
arruelas anilhas incluindo pressao outros artigos roscados tachas pr
egos percevejos escapulas artigos semelhantes cobre ferro cabeca cob
re parafusos pinos pernos roscados porcas ganchos roscados rebites c
havetas cavilhas contrapinos trocos arruelas anilhas incluindo press
ao artigos semelhantes cobre
0.30059666002668056
2
outros outros artigos roscados tachas pregos percevejos escapulas ar
tigos semelhantes cobre ferro cabeca cobre parafusos pinos pernos ro
scados porcas ganchos roscados rebites chavetas cavilhas contrapinos
trocos arruelas anilhas incluindo pressao artigos semelhantes cobre
0.1668534404185063
1
parafusos pinos pernos porcas outros artigos roscados tachas pregos
percevejos escapulas artigos semelhantes cobre ferro cabeca cobre pa
rafusos pinos pernos roscados porcas ganchos roscados rebites chavet
as cavilhas contrapinos trocos arruelas anilhas incluindo pressao ar
tigos semelhantes cobre
0.14578626546105106
1
outros outros artigos roscados tachas pregos percevejos escapulas ar
tigos semelhantes cobre ferro cabeca cobre parafusos pinos pernos ro
scados porcas ganchos roscados rebites chavetas cavilhas contrapinos
trocos arruelas anilhas incluindo pressao artigos semelhantes cobre
0.1668534404185063
1
tachas pregos escapulas parafusos pinos pernos roscados porcas ganch
os roscados rebites chavetas cavilhas contrapinos trocos arruelas an
ilhas artigos semelhantes outras obras aluminio
0.2195719174909269
1

In [24]:

```
show_dados_palavra(listaTECResumo, X, Xtfid, 'arruelas')
```

3926.90.10 Arruelas - Outras Outras obras de plástico e obras de o
utras matérias das posições 39.01 a 39.14.
0.35209044782641996
1
4006.10.00 - Perfis para recauchutagem Outras formas (por exempl
o, varetas, tubos, perfis) e artigos (por exemplo, discos, arruelas
(anilhas*)), de borracha não vulcanizada.
0.2874789688267496
1
4006.90.00 - Outros Outras formas (por exemplo, varetas, tubos, p
erfis) e artigos (por exemplo, discos, arruelas (anilhas*)), de borr
acha não vulcanizada.
0.3380702594906789
1
7318.11.00 -- Tira-fundos - Artigos roscados: Parafusos, pinos o
u pernos, roscados, porcas, tira-fundos, ganchos roscados, rebites,
chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*) (incl
uindo as de pressão) e artigos semelhantes, de ferro fundido, ferro
ou aço.
0.17171849214498303
1
7318.12.00 -- Outros parafusos para madeira - Artigos roscados: Pa
rafusos, pinos ou pernos, roscados, porcas, tira-fundos, ganchos ros
cados, rebites, chavetas, cavilhas, contrapinos ou troços, arruelas
(anilhas*) (incluindo as de pressão) e artigos semelhantes, de ferro
fundido, ferro ou aço.
0.18128620627348382
1
7318.13.00 -- Ganchos e armelas (pitões*) - Artigos roscados: Pa
rafusos, pinos ou pernos, roscados, porcas, tira-fundos, ganchos ros
cados, rebites, chavetas, cavilhas, contrapinos ou troços, arruelas
(anilhas*) (incluindo as de pressão) e artigos semelhantes, de ferro
fundido, ferro ou aço.
0.1720059105961564
1
7318.14.00 -- Parafusos perfurantes - Artigos roscados: Parafusos,
pinos ou pernos, roscados, porcas, tira-fundos, ganchos roscados, re
bites, chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas
*) (incluindo as de pressão) e artigos semelhantes, de ferro fundid
o, ferro ou aço.
0.1776287014679304
1
7318.15.00 -- Outros parafusos e pinos ou pernos, mesmo com as por
cas e arruelas (anilhas*) - Artigos roscados: Parafusos, pinos o
u pernos, roscados, porcas, tira-fundos, ganchos roscados, rebites,
chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*) (incl
uindo as de pressão) e artigos semelhantes, de ferro fundido, ferro
ou aço.
0.2996100476868767
2
7318.16.00 -- Porcas - Artigos roscados: Parafusos, pinos o
u pernos, roscados, porcas, tira-fundos, ganchos roscados, rebites,
chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*) (incl
uindo as de pressão) e artigos semelhantes, de ferro fundido, ferro
ou aço.
0.1827727597519414
1
7318.19.00 -- Outros - Artigos roscados: Parafusos, pinos o
u pernos, roscados, porcas, tira-fundos, ganchos roscados, rebites,
chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*) (incl
uindo as de pressão) e artigos semelhantes, de ferro fundido, ferro

ou aço.
0.1916006548870102
1
7318.21.00 -- Arruelas (Anilhas*) de pressão e outras arruelas (an
ilhas*) de segurança - Artigos não roscados: Parafusos, pinos ou pe
rnos, roscados, porcas, tira-fundos, ganchos roscados, rebites, chav
etas, cavilhas, contrapinos ou troços, arruelas (anilhas*) (incluind
o as de pressão) e artigos semelhantes, de ferro fundido, ferro ou a
ço.
0.44070400092879897
3
7318.22.00 -- Outras arruelas (anilhas*) - Artigos não roscado
s: Parafusos, pinos ou pernos, roscados, porcas, tira-fundos, gancho
s roscados, rebites, chavetas, cavilhas, contrapinos ou troços, arru
elas (anilhas*) (incluindo as de pressão) e artigos semelhantes, de
ferro fundido, ferro ou aço.
0.3463490721909246
2
7318.23.00 -- Rebites - Artigos não roscados: Parafusos, pin
os ou pernos, roscados, porcas, tira-fundos, ganchos roscados, rebit
es, chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*)
(incluindo as de pressão) e artigos semelhantes, de ferro fundido, f
erro ou aço.
0.1826026978901023
1
7318.24.00 -- Chavetas, cavilhas e contrapinos ou troços - Arti
gos não roscados: Parafusos, pinos ou pernos, roscados, porcas, tira
-fundos, ganchos roscados, rebites, chavetas, cavilhas, contrapinos
ou troços, arruelas (anilhas*) (incluindo as de pressão) e artigos s
emelhantes, de ferro fundido, ferro ou aço.
0.15873355931187477
1
7318.29.00 -- Outros - Artigos não roscados: Parafusos, pin
os ou pernos, roscados, porcas, tira-fundos, ganchos roscados, rebit
es, chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*)
(incluindo as de pressão) e artigos semelhantes, de ferro fundido, f
erro ou aço.
0.1916006548870102
1
7415.10.00 - Tachas, pregos, percevejos, escápulas e artigos seme
lhantes Tachas, pregos, percevejos, escápulas e artigos semelhantes,
de cobre ou de ferro ou aço com cabeça de cobre; parafusos, pinos ou
pernos, roscados, porcas, ganchos roscados, rebites, chavetas, cavil
has, contrapinos ou troços, arruelas (anilhas*) (incluindo as de pre
ssão), e artigos semelhantes, de cobre.
0.14737616600525336
1
7415.21.00 -- Arruelas (Anilhas*) (incluindo as de pressão) -
Outros artigos, não roscados: Tachas, pregos, percevejos, escápulas
e artigos semelhantes, de cobre ou de ferro ou aço com cabeça de cob
re; parafusos, pinos ou pernos, roscados, porcas, ganchos roscados,
rebites, chavetas, cavilhas, contrapinos ou troços, arruelas (anilha
s*) (incluindo as de pressão), e artigos semelhantes, de cobre.
0.30059666002668056
2
7415.29.00 -- Outros - Outros artigos, não roscados: Tacha
s, pregos, percevejos, escápulas e artigos semelhantes, de cobre ou
de ferro ou aço com cabeça de cobre; parafusos, pinos ou pernos, ros
cados, porcas, ganchos roscados, rebites, chavetas, cavilhas, contra
pinos ou troços, arruelas (anilhas*) (incluindo as de pressão), e ar
tigos semelhantes, de cobre.

0.1668534404185063
1
7415.33.00 -- Parafusos; pinos ou pernos e porcas - Outros artigos, roscados: Tachas, pregos, percevejos, escápulas e artigos semelhantes, de cobre ou de ferro ou aço com cabeça de cobre; parafusos, pinos ou pernos, roscados, porcas, ganchos roscados, rebites, chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*) (incluindo as de pressão), e artigos semelhantes, de cobre.
0.14578626546105106
1
7415.39.00 -- Outros - Outros artigos, roscados: Tachas, pregos, percevejos, escápulas e artigos semelhantes, de cobre ou de ferro ou aço com cabeça de cobre; parafusos, pinos ou pernos, roscados, porcas, ganchos roscados, rebites, chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*) (incluindo as de pressão), e artigos semelhantes, de cobre.
0.1668534404185063
1
7616.10.00 - Tachas, pregos, escápulas, parafusos, pinos ou pernos roscados, porcas, ganchos roscados, rebites, chavetas, cavilhas, contrapinos ou troços, arruelas (anilhas*) e artigos semelhantes Outras obras de alumínio.
0.2195719174909269
1

In [25]:

```
show_dados_palavra(listaTECResumo, X, Xtfid, 'bolsas')
```

3006.91.10 Bolsas para colostomia, ileostomia e urostomia -- Equi
pamentos identificáveis para ostomia - Outros: Preparações e artigo
s farmacêuticos indicados na Nota 4 deste Capítulo.
0.25011218243894123
1
3923.21.10 De capacidade inferior ou igual a 1.000 cm3 -- De p
olímeros de etileno - Sacos de quaisquer dimensões, bolsas e cartu
chos: Artigos de transporte ou de embalagem, de plástico; rolhas, ta
mpas, cápsulas e outros dispositivos para fechar recipientes, de plá
stico.
0.23243558246964305
1
3923.21.90 Outros -- De polímeros de etileno - Sacos de qua
isquer dimensões, bolsas e cartuchos: Artigos de transporte ou de em
balagem, de plástico; rolhas, tampas, cápsulas e outros dispositivos
para fechar recipientes, de plástico.
0.2530928044598255
1
3923.29.10 De capacidade inferior ou igual a 1.000 cm3 -- De o
utro plástico - Sacos de quaisquer dimensões, bolsas e cartuchos: Ar
tigos de transporte ou de embalagem, de plástico; rolhas, tampas, cá
psulas e outros dispositivos para fechar recipientes, de plástico.
0.22533839354086785
1
3923.29.90 Outros -- De outro plástico - Sacos de quaisquer d
imensões, bolsas e cartuchos: Artigos de transporte ou de embalagem,
de plástico; rolhas, tampas, cápsulas e outros dispositivos para fec
har recipientes, de plástico.
0.24400662282044192
1
3926.90.30 Bolsas para uso em medicina (hemodiálise e usos semelhant
es) - Outras Outras obras de plástico e obras de outras matérias d
as posições 39.01 a 39.14.
0.2868054531339832
1
4014.90.10 Bolsas para gelo ou para água quente - Outros Artig
os de higiene ou de farmácia (incluindo as chupetas), de borracha vu
lcanizada não endurecida, mesmo com partes de borracha endurecida.
0.2550210499465337
1
4202.11.00 -- Com a superfície exterior de couro natural ou recons
tituído - Baús (Arcas*) para viagem, malas e maletas, incluind
o as maletas de toucador e as maletas e pastas de documentos e para
estudantes, e artigos semelhantes: Baús (Arcas*) para viagem, malas
e maletas, incluindo as maletas de toucador e as maletas e pastas de
documentos e para estudantes, os estojos para óculos, binóculos, câm
eras fotográficas e de filmar, instrumentos musicais, armas e artigo
s semelhantes; sacos de viagem, sacos isolantes para gêneros aliment
ícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.18310101277335428
2
4202.12.10 De plástico -- Com a superfície exterior de plástic
o ou de matérias têxteis - Baús (Arcas*) para viagem, malas e m
aletas, incluindo as maletas de toucador e as maletas e pastas de do
cumentos e para estudantes, e artigos semelhantes: Baús (Arcas*) par
a viagem, malas e maletas, incluindo as maletas de toucador e as mal
etas e pastas de documentos e para estudantes, os estojos para óculo
s, binóculos, câmeras fotográficas e de filmar, instrumentos musicai
s, armas e artigos semelhantes; sacos de viagem, sacos isolantes par
a gêneros alimentícios e bebidas, bolsas de toucador, mochilas, bols
as,

0.1831291217813744
2
4202.12.20 De matérias têxteis -- Com a superfície exterior de plástico ou de matérias têxteis - Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, e artigos semelhantes: Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, os estojos para óculos, binóculos, câmeras fotográficas e de filmar, instrumentos musicais, armas e artigos semelhantes; sacos de viagem, sacos isolantes para gêneros alimentícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.1825897728969143
2
4202.19.00 -- Outros - Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, e artigos semelhantes: Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, os estojos para óculos, binóculos, câmeras fotográficas e de filmar, instrumentos musicais, armas e artigos semelhantes; sacos de viagem, sacos isolantes para gêneros alimentícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.18674551454964258
2
4202.21.00 -- Com a superfície exterior de couro natural ou reconstituído - Bolsas, mesmo com tiracolo, incluindo as que não possuam alças (pegas*): Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, os estojos para óculos, binóculos, câmeras fotográficas e de filmar, instrumentos musicais, armas e artigos semelhantes; sacos de viagem, sacos isolantes para gêneros alimentícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.3711807130900996
3
4202.22.10 De folhas de plástico -- Com a superfície exterior de folhas de plástico ou de matérias têxteis - Bolsas, mesmo com tiracolo, incluindo as que não possuam alças (pegas*): Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, os estojos para óculos, binóculos, câmeras fotográficas e de filmar, instrumentos musicais, armas e artigos semelhantes; sacos de viagem, sacos isolantes para gêneros alimentícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.36661671206212093
3
4202.22.20 De matérias têxteis -- Com a superfície exterior de folhas de plástico ou de matérias têxteis - Bolsas, mesmo com tiracolo, incluindo as que não possuam alças (pegas*): Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, os estojos para óculos, binóculos, câmeras fotográficas e de filmar, instrumentos musicais, armas e artigos semelhantes; sacos de viagem, sacos isolantes para gêneros alimentícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.3681294939368213
3
4202.29.00 -- Outras - Bolsas, mesmo com tiracolo, incluindo as que não possuam alças (pegas*): Baús (Arcas*) para viagem, malas e maletas, incluindo as maletas de toucador e as maletas e pastas de documentos e para estudantes, os estojos para óculos, binóculos,

câmeras fotográficas e de filmar, instrumentos musicais, armas e art
igos semelhantes; sacos de viagem, sacos isolantes para gêneros alim
entícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.38477849178217677
3
4202.31.00 -- Com a superfície exterior de couro natural ou recons
tituído - Artigos do tipo normalmente levado nos bolsos ou em
bolsas: Baús (Arcas*) para viagem, malas e maletas, incluindo as mal
etas de toucador e as maletas e pastas de documentos e para estudant
es, os estojos para óculos, binóculos, câmeras fotográficas e de fil
mar, instrumentos musicais, armas e artigos semelhantes; sacos de vi
agem, sacos isolantes para gêneros alimentícios e bebidas, bolsas de
toucador, mochilas, bolsas,
0.37382036654714934
3
4202.32.00 -- Com a superfície exterior de folhas de plástico ou d
e matérias têxteis - Artigos do tipo normalmente levado nos bolso
s ou em bolsas: Baús (Arcas*) para viagem, malas e maletas, incluind
o as maletas de toucador e as maletas e pastas de documentos e para
estudantes, os estojos para óculos, binóculos, câmeras fotográficas
e de filmar, instrumentos musicais, armas e artigos semelhantes; sac
os de viagem, sacos isolantes para gêneros alimentícios e bebidas, b
olsas de toucador, mochilas, bolsas,
0.37672849945360765
3
4202.39.00 -- Outros - Artigos do tipo normalmente levado n
os bolsos ou em bolsas: Baús (Arcas*) para viagem, malas e maletas,
incluindo as maletas de toucador e as maletas e pastas de documentos
e para estudantes, os estojos para óculos, binóculos, câmeras fotogr
áficas e de filmar, instrumentos musicais, armas e artigos semelhant
es; sacos de viagem, sacos isolantes para gêneros alimentícios e beb
idas, bolsas de toucador, mochilas, bolsas,
0.38796782556762244
3
4202.91.00 -- Com a superfície exterior de couro natural ou recons
tituído - Outros: Baús (Arcas*) para viagem, malas e maletas,
incluindo as maletas de toucador e as maletas e pastas de documentos
e para estudantes, os estojos para óculos, binóculos, câmeras fotogr
áficas e de filmar, instrumentos musicais, armas e artigos semelhant
es; sacos de viagem, sacos isolantes para gêneros alimentícios e beb
idas, bolsas de toucador, mochilas, bolsas,
0.27418341977895355
2
4202.92.00 -- Com a superfície exterior de folhas de plástico ou d
e matérias têxteis - Outros: Baús (Arcas*) para viagem, malas e m
aletas, incluindo as maletas de toucador e as maletas e pastas de do
cumentos e para estudantes, os estojos para óculos, binóculos, câmer
as fotográficas e de filmar, instrumentos musicais, armas e artigos
semelhantes; sacos de viagem, sacos isolantes para gêneros alimentíc
ios e bebidas, bolsas de toucador, mochilas, bolsas,
0.27677164618153915
2
4202.99.00 -- Outros - Outros: Baús (Arcas*) para viagem, m
alas e maletas, incluindo as maletas de toucador e as maletas e past
as de documentos e para estudantes, os estojos para óculos, binóculo
s, câmeras fotográficas e de filmar, instrumentos musicais, armas e
artigos semelhantes; sacos de viagem, sacos isolantes para gêneros a
limentícios e bebidas, bolsas de toucador, mochilas, bolsas,
0.2865583038939943
2
4819.10.00 - Caixas de papel ou cartão, ondulados (canelados*) Ca

ixas, sacos, bolsas, cartuchos e outras embalagens, de papel, cartã
o, pasta (ouate) de celulose ou de mantas de fibras de celulose; car
tonagens para escritórios, lojas e estabelecimentos semelhantes.
0.20340602363223614
1
4819.20.00 - Caixas e cartonagens, dobráveis, de papel ou cartão,
não ondulados (não canelados*) Caixas, sacos, bolsas, cartuchos e ou
tras embalagens, de papel, cartão, pasta (ouate) de celulose ou de m
antas de fibras de celulose; cartonagens para escritórios, lojas e e
stabelecimentos semelhantes.
0.17948281934870097
1
4819.30.00 - Sacos cuja base tenha largura igual ou superior a 4
0 cm Caixas, sacos, bolsas, cartuchos e outras embalagens, de papel,
cartão, pasta (ouate) de celulose ou de mantas de fibras de celulos
e; cartonagens para escritórios, lojas e estabelecimentos semelhante
s.
0.20841288905964692
1
4819.40.00 - Outros sacos; bolsas e cartuchos Caixas, sacos, bols
as, cartuchos e outras embalagens, de papel, cartão, pasta (ouate) d
e celulose ou de mantas de fibras de celulose; cartonagens para escr
itórios, lojas e estabelecimentos semelhantes.
0.3971233899250121
2
4819.50.00 - Outras embalagens, incluindo as capas para discos Ca
ixas, sacos, bolsas, cartuchos e outras embalagens, de papel, cartã
o, pasta (ouate) de celulose ou de mantas de fibras de celulose; car
tonagens para escritórios, lojas e estabelecimentos semelhantes.
0.21768328897376601
1
4819.60.00 - Cartonagens para escritórios, lojas e estabeleciment
os semelhantes Caixas, sacos, bolsas, cartuchos e outras embalagens,
de papel, cartão, pasta (ouate) de celulose ou de mantas de fibras d
e celulose; cartonagens para escritórios, lojas e estabelecimentos s
emelhantes.
0.1671132917277739
1
8484.10.00 - Juntas metaloplásticas Juntas metaloplásticas; jogos
ou sortidos de juntas de composições diferentes, apresentados em bol
sas, envelopes ou embalagens semelhantes; juntas de vedação mecânica
s.
0.17875785545987
1
8484.20.00 - Juntas de vedação mecânicas Juntas metaloplásticas;
jogos ou sortidos de juntas de composições diferentes, apresentados
em bolsas, envelopes ou embalagens semelhantes; juntas de vedação me
cânicas.
0.17333498498406594
1
8484.90.00 - Outros Juntas metaloplásticas; jogos ou sortidos de
juntas de composições diferentes, apresentados em bolsas, envelopes
ou embalagens semelhantes; juntas de vedação mecânicas.
0.2243513700300032
1
8708.95.10 Bolsas infláveis de segurança com sistema de insuflação
(airbags) -- Bolsas infláveis de segurança com sistema de insufla
ção (airbags); suas partes - Outras partes e acessórios: Partes e
acessórios dos veículos automóveis das posições 87.01 a 87.05.
0.3184266484054504
2

```
8708.95.21 Bolsas infláveis para airbags Partes --      Bolsas inflá
veis de segurança com sistema de insuflação (airbags); suas partes -
Outras partes e acessórios: Partes e acessórios dos veículos automóv
eis das posições 87.01 a 87.05.
0.35880073399867685
2
8708.95.22 Sistema de insuflação Partes --      Bolsas infláveis de
segurança com sistema de insuflação (airbags); suas partes -    Outr
as partes e acessórios: Partes e acessórios dos veículos automóveis
das posições 87.01 a 87.05.
0.19532471055666212
1
8708.95.29 Outras Partes --     Bolsas infláveis de segurança com si
stema de insuflação (airbags); suas partes -    Outras partes e aces
sórios: Partes e acessórios dos veículos automóveis das posições 87.
01 a 87.05.
0.2278822607548185
1
9018.32.12 De aço cromo-níquel, bisel trifacetado e diâmetro exterio
r igual ou superior a 1,6 mm, do tipo das utilizadas com bolsas de s
angue Tubulares de metal --     Agulhas tubulares de metal e agulhas
para suturas -  Seringas, agulhas, cateteres, cânulas e instrumentos
semelhantes: Instrumentos e aparelhos para medicina, cirurgia, odont
ologia e veterinária, incluindo os aparelhos para cintilografia e ou
tros aparelhos eletromédicos, bem como os aparelhos para testes visu
ais.
0.15047450950119262
1
```

Conclusão TFIDF básico

Conforme demonstrado acima, é possível usar a matriz TFIDF para fazer "buscas" de classificação fiscal na TEC. Bastaria consultar os valores TFIDF de cada palavra, combinar, e exibir o "documento" (subitem da TEC) na ordem da pontuação obtida.

Teste classificador (é possível predizer propabilidade de capítulo pela frase??)

In [26]:

```
y = [int(linha[:2]) for linha in listaTECResumo]
y
```

Out[26]:

```
[1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
```

```
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
3,
```

```
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
```

```
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
```

```
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
```

```
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
```

```
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
```

```
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
4,
```

```
    4,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    5,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    6,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
```

```
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
    7,
```

```
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
7,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
```

```
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
8,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
9,
```

```
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    9,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
    10,
```

```
10,
10,
10,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
11,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
```

```
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
12,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
13,
14,
```

```
14,
14,
14,
14,
14,
14,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
15,
```

```
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 15,
 16,
 16,
 16,
 16,
 16,
 16,
 16,
 16,
 ...]
```

In [27]:

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
```

In [28]:

```
X_train, X_val, y_train, y_val = train_test_split(X, y, train_size=0.2)
clf = MultinomialNB()
clf.fit(X_train, y_train)
clf.score(X_val, y_val)
```

Out[28]:

```
0.8223700418822371
```

In [29]:

```
arruela_plastico = vectorizer.transform(['arruelas plastico'])
print(arruela_plastico)
```

```
  (0, 936)      1
  (0, 7102)     1
```

In [30]:

```
clf.predict(arruela_plastico)
```

Out[30]:

```
array([39])
```

In [31]:

```
bolsas_couro = vectorizer.transform(['bolsas couro'])
clf.predict(bolsas_couro)
```

Out[31]:

```
array([38])
```

In [32]:

```
bolsas_plastico = vectorizer.transform(['bolsas plastico'])
clf.predict(bolsas_plastico)
```

Out[32]:

```
array([39])
```

In [33]:

```
X_train, X_val, y_train, y_val = train_test_split(Xtfid, y, train_size=0.2)
clf = MultinomialNB()
clf.fit(X_train, y_train)
clf.score(X_val, y_val)
```

Out[33]:

```
0.5700911554570092
```

Melhorias

A grande fragilidade destas possibilidades de busca demonstradas (usar TF-IDF ou treinar classificadores) é que as palavras buscadas **precisam** estar no vocabulário. Para suprir este problema, são possíveis duas abordagens:

- Usar um dicionário de sinônimos
- Usar word embeddings e treinar um classificador com as embeddings (há um problema, como a base tem somente pouco mais de 10.000 linhas e este método envolve redes neurais, provavelmente não será possível treinar redes neurais)

In [13]:

```
import spacy
nlp = spacy.load("pt_core_news_sm")

tokens = nlp('arruelas plastico')
```

In [35]:

```
tokens
```

Out[35]:

```
arruelas plastico
```

In [36]:

```
for token in tokens:
    print(token.text, token.has_vector, token.vector_norm, token.is_oov)
```

```
arruelas True 48.852642 True
plastico True 44.635536 True
```

In [37]:

```
tokens[0].vector
```

Out[37]:

```
array([ 12.776641  ,  -1.119842  ,   4.2651024 ,  -2.842749  ,
         3.3195808 ,   2.4608216 ,  -8.104986  ,  -3.493442  ,
         5.909817  ,   0.86661834,   4.5411987 ,   1.7983334 ,
        -1.0091991 ,   2.2095509 ,  -5.637903  ,   5.109632  ,
        -2.7778745 ,  -3.3918538 ,  -2.3148997 ,   3.998908  ,
         8.419077  ,   1.0418421 ,   4.046485  ,  11.753028  ,
         1.8997545 ,  -4.8082795 ,  -5.912253  ,  -4.211125  ,
         2.3638601 ,   1.2602189 ,   2.5386271 ,  -5.6300163 ,
         5.386259  ,  -6.4208493 ,  -2.320203  ,  -3.84236   ,
         5.898923  ,   1.9341607 ,  -3.6678014 ,   2.350322  ,
        -3.1752608 ,   8.008952  ,   6.3870726 ,   1.5020604 ,
        -7.114374  ,  -3.8284023 ,  -4.3331447 ,   1.3941951 ,
         5.91114   ,   1.3393854 ,  -4.2787294 ,  -6.9610586 ,
        -3.0004895 ,  -2.647688  ,   1.2795722 ,  -1.1136105 ,
         0.6729131 ,   0.30478424,  -5.09515   ,   5.5853558 ,
        -5.595108  ,  -1.9178748 ,  -1.5372154 ,  -4.701381  ,
        -2.0815692 ,  -5.5126853 ,   0.5369953 ,   0.5162421 ,
         1.7840258 ,  -1.6818159 ,  11.1459    ,  -4.516899  ,
        -2.5338404 ,   9.547797  ,  10.50752   ,   0.06267428,
        -8.231815  ,  -1.3078612 ,   1.9740145 ,   2.9392958 ,
         2.358956  ,   4.001782  ,  -4.0522757 ,   3.1960907 ,
        -8.1803    ,   0.55904496,  -5.1933403 ,  -2.9904675 ,
        10.5393505 ,   2.5870814 ,   2.444601  ,  -3.1266541 ,
       -14.229488  ,  -1.7127447 ,  -2.6502488 ,   4.667042  ],
      dtype=float32)
```

In [38]:

```
tokens[1].vector
```

Out[38]:

```
array([  3.5684836 ,  -1.925127  ,   2.548791  ,   2.1013541 ,
         6.9125543 ,  -1.157139  ,  -6.622593  ,  -7.2240734 ,
        -1.7155715 ,   3.7684174 ,   6.43216   ,   1.0183662 ,
        -2.4061189 ,   0.11474317,   2.5214686 ,   2.13903   ,
        -3.5432763 ,   8.414198  ,  -1.1926565 ,   4.780589  ,
         3.593836  ,  -0.07527542,   8.304298  ,  11.163615  ,
        -3.867725  ,  -1.0414963 ,  -1.797202  ,   1.5334928 ,
         0.6240573 ,   0.5808499 , -10.1132    ,  -5.670818  ,
        -2.4708009 ,   7.4862423 ,   5.04297   ,  -1.0860665 ,
        -3.620214  ,  -7.5375605 ,  -1.2403668 ,   5.463876  ,
        -0.25288314,  -1.9830638 ,  -0.19251204,   4.277309  ,
        -3.1096148 ,   3.8140502 ,  -8.031469  ,   3.5311406 ,
        -2.001999  ,  -0.16415465,  -3.211752  ,  -2.2337608 ,
        -2.0819087 ,  11.009416  ,   5.780795  ,  -3.5985699 ,
        -6.363167  ,   7.7743864 ,   5.826233  ,   1.4261215 ,
        -0.29241854,  -0.7948784 ,  -3.2691932 ,  -5.434671  ,
         5.1389194 ,  -2.0669408 ,   3.3211598 ,  -4.852355  ,
        -1.1071029 ,  -2.0985026 ,   5.9485736 ,  -7.3796844 ,
         0.9722601 ,   0.3029228 ,  -3.652341  ,   1.2458359 ,
        -2.348308  ,   0.5216124 ,  -0.34837905,   0.6642592 ,
        -1.5336947 ,  -3.8738973 ,  -4.1033325 ,   0.20987916,
         1.7767528 ,   3.6118567 , -11.222702  ,   1.519663  ,
         1.7581381 ,   4.662358  ,   0.13999763,   2.3428621 ,
       -12.286081  ,   1.5987071 ,   4.7821493 ,  -4.6349716 ],
      dtype=float32)
```

In [39]:

```
from tensorflow import keras
```

In [40]:

```
for doc in corpus[:2]:
    print(doc)
    tokens = nlp(doc)
    print(tokens.vector)
```

```
reprodutores raca pura cavalos cavalos asininos muares vivos
[ 2.7953186   4.176491    0.8434681   0.16618407  2.6092045  -1.1901
851
  1.2190241  -3.5989938   2.6585293  -0.5147271   1.0065104  -0.2972
5927
  4.733627    4.3019066  -0.15027341  3.0099165  -0.5224895   3.0358
  2.7201717   2.8244414   5.4925346  -2.0034776   1.8145082   3.3846
107
 -4.968304   -4.7737427  -0.25154695 -0.31632626 -2.104146   -1.9016
676
 -3.6670713  -6.180348    3.940358    2.0538275   3.5724587  -4.4996
57
  5.1532817  -4.3737025  -1.5316811  -5.238574    0.13850509  0.5397
528
  0.2061348  -1.8139625  -2.1200364  -4.512215   -4.2306376   2.2479
71
 -1.7034007  -3.4776506   1.3487179  -7.4980073   2.0507877   2.0126
24
  4.8183413  -0.7075846   9.25494     0.6818701   4.022955   -0.9694
898
 -1.1424521  -3.6532862  -1.9534218  -7.028407    7.2797275  -5.3349
67
  3.8741927   2.8014352   2.993805   -0.23742156  4.7158194  -5.1164
284
 -3.7728276   1.474102   -2.6845472  -2.0615275   2.2942512  -4.4975
15
 -0.73931825  0.19035333  2.2980773  -1.2301376  -2.7103686  -2.5607
603
 -2.6825895   7.5659766  -4.399836   -2.3841858   2.6325874   3.9022
83
  7.4638405  -0.75525284 -3.756831   -1.4950943   1.5254508   0.3679
9842]
outros cavalos cavalos asininos muares vivos
[ 4.0672174   7.6592956  -0.5800944   1.813825    1.7103728  -0.4874
6035
  1.3230141  -5.23872     4.967605    1.5160608   2.8934963   0.3111
5422
  4.6651177   7.1863346  -0.69644684  3.4309103  -0.5640103   3.7222
52
  1.9856591   2.5590665   2.918457   -2.073738    0.7338309   1.4388
676
 -3.5775592  -7.2148786  -2.0590782  -2.4862983  -1.8398839  -1.6259
729
 -3.2517278  -5.5891786   3.6498146   2.5155303   2.979043   -4.7213
206
  3.449314   -6.487356   -3.405219   -5.6765695   0.8912696   0.9977
183
  1.0769116  -3.9899437  -1.7250932  -5.8881125  -4.136043    5.3373
084
 -2.10619    -2.3918164   3.4281452  -8.315402    1.4195746   1.9465
866
  6.1738563  -1.4775285  10.176648    0.12464619  4.0536714  -0.5229
2794
 -0.88719344 -4.4708896  -4.357105   -7.8357944   5.9360414  -4.1058
47
  3.183026    1.3968878   3.2245715   0.67785054  5.9234543  -4.9771
67
 -4.1779075   2.463231   -1.845698   -1.9942936   0.6486909  -5.2175
58
 -3.7141855   1.0364406   2.4434278   2.2770765  -3.3883572  -3.9101
01
```

```
    -3.0987918   9.140574   -2.115234   -2.5495422   1.5823878   2.4332
952
     9.4635      1.0594248  -3.1113675  -3.3444307  -1.1752436   3.3153
353 ]
```

In [41]:

```
for doc in corpus[:2]:
    print(doc)
    tokens = nlp(doc)
    for token in tokens:
        print(token.vector)
```

```
  reprodutores raca pura cavalos cavalos asininos muares vivos
  [ 6.64293242e+00  1.87291443e-01  6.24938774e+00  1.25831628e+00
    1.12645035e+01  1.62478197e+00 -6.70816994e+00 -1.23039436e+00
    5.05357218e+00 -3.12805176e+00  1.66952193e+00 -3.97077203e+00
    4.87114954e+00  7.86261654e+00 -6.47044659e+00  1.94734764e+00
    2.06741238e+00  6.61689878e-01  3.94642997e+00 -1.63302869e-02
    6.39751863e+00 -3.85731220e-01 -5.53474426e-01  3.19521689e+00
    1.31331623e-01 -1.11653461e+01 -1.57218754e+00 -5.52335083e-01
    1.86404061e+00 -2.52212238e+00 -4.66619825e+00 -7.23742867e+00
    8.59963608e+00  1.08069718e+00  6.40867996e+00 -8.20451927e+00
    5.25842619e+00 -9.63336372e+00 -2.79402876e+00  1.02870238e+00
    1.69568944e+00  4.14215183e+00 -3.32162738e+00  4.07819867e-01
   -6.81309938e+00 -9.28379536e+00 -2.13685131e+00  7.57637453e+00
    4.28763151e+00  3.81842184e+00 -5.71094632e-01 -1.06393709e+01
   -8.56875801e+00 -5.78028321e-01  5.24746656e+00  3.46253490e+00
    1.76879239e+00 -4.92994595e+00 -2.56430328e-01 -5.36544442e-01
   -6.48282909e+00 -1.30999970e+00 -8.14078450e-01 -6.34335136e+00
    4.05887890e+00 -3.61303830e+00  1.39137745e+00  2.30554628e+00
    1.39408684e+00 -2.85495901e+00  1.12897825e+01 -1.98387003e+00
    2.19006634e+00 -3.19921970e-03  4.93487692e+00 -2.75957131e+00
    5.29434919e+00 -4.04155397e+00 -2.30014086e+00  1.68976343e+00
   -1.23612642e-01  4.63714743e+00 -3.70339537e+00  2.06048942e+00
   -6.55541229e+00  4.76594925e+00 -6.29005575e+00  4.88093406e-01
    7.20721579e+00 -2.39367485e+00  1.37540264e+01 -6.44063854e+00
   -7.21631908e+00 -3.75078368e+00  5.73268771e-01  3.10407495e+00]
  [  1.0863048   -2.7894533    6.9680557    0.08295405   8.480454
    -0.23761934  -0.68354964  -5.7243595   -5.0674005    4.1179113
     2.51198     -5.6980796   -0.24173518  -3.6367745    4.9928293
     6.2197914    2.15706     -4.435322     4.0617003    0.9087716
     4.7484436    2.668272     6.628061     7.500891    -3.9301577
    -8.1157875    3.061476     1.6057082    0.2058376   -4.239825
    -7.7303863   -4.3014393    7.102921     4.908053     6.276469
    -3.8917956    5.7583046   -1.0862677    6.300334     0.75612783
    -4.88805     -0.766531    -3.4806933    3.9001112   -5.8072376
    -0.11451912  -1.4290804    6.4062853   -6.078262    -2.2355113
    -3.9690788   -6.9767547    2.033164     3.0988002    2.6800663
     4.3491077   -0.18103781   4.9755144    1.0837207   -6.5856857
    -6.095976     0.3742739    3.2784503   -3.587216    11.426226
    -2.8507655   -1.9860098    4.735774     1.7975597   -0.62502664
    -4.2821817   -4.464139    -4.9245424   -8.896518     0.8314903
    -0.251832     6.267582    -0.5932001    2.478343     2.056869
     7.0361414   -6.071352    -4.1507573    3.7893767   -1.1372217
     3.4948878   -6.7170315   -3.7457006    3.6472187    9.098037
     0.47309887  -3.617497   -12.048483     0.0188278    3.887532
    -6.6303854 ]
  [  2.1992743    8.091729     4.522311    -8.188901    -7.0303016
     0.45658055   4.826764    -3.585285    -7.113357    -3.4468498
    -2.8221853   -3.3022718    1.9646587   -2.1140697   -9.23862
     6.680383    -4.5580435    2.3379543    4.260918     5.518238
     5.128637     3.8368454    6.381139     5.4880304  -12.824165
     1.170821     3.2437282   -2.5353253   -0.12281916   3.536547
     3.3140566   -7.2784824    1.7457491   -0.7575569    0.47740614
     4.1811776    5.8105583    1.6979222   -3.199921    -7.532384
    -3.2352204   -1.7950699   -3.8054972   -2.0748372   -0.7112931
     7.410285    -3.1889167   -7.5247235   -0.6689456   -4.497854
    -0.52249455  -6.6036496    9.145586    -0.80700564   1.5849553
    -2.0926564   11.816529     4.875844     3.7209067    3.7519534
     1.0406731   -2.5321631    4.894824    -0.89243233  -0.5185905
    -7.3550806    9.400541     3.9300883   -0.54305255  -5.671421
     1.1497175   -2.7825446   -5.751813    -2.8366346   -3.3904161
     2.5586598    0.456926    -3.3449183    6.023432    -0.99129224
```

```
     4.262294    -4.4304085   -2.8050134   -2.4955997   -1.8132106
     4.870413    -1.4493029    1.2932534    9.012482     5.160825
     0.860921    -7.857154     1.8112178    2.6701531    1.7830403
    -4.2964616 ]
  [ -0.73945105   2.8385053    1.0226985   -0.8699119    0.52795815
    -1.3864001    4.200994    -0.83615595   6.2351303   -5.654449
    -5.499399    -0.74021846   8.0443325   10.607567     0.6549323
    -0.64773285  -3.5004978    7.270165     2.1535282    2.1119525
     7.592078    -3.0291953   -4.9867024    6.358389    -5.504796
     0.8302175   -2.2097998   -0.89226305  -5.209181     2.8009102
    -0.6648627   -5.170225     2.3313642   -1.2005141   -0.17824328
   -10.282638     5.1335278   -4.600122    -5.960139    -7.4703474
     2.422846     0.66232747   1.0053172   -0.69547033  -2.6582355
    -5.127459    -5.0771008    0.898503     1.7458392   -4.2765102
     0.84111696  -4.7694664    4.3065453    0.45421484   5.332529
    -0.11407435  10.148493    -3.6055446    3.6794648   -1.1967767
     8.061204    -4.457557    -1.1100774   -9.182857    10.561043
   -10.291653     7.2363005    0.9260359    2.4826021    5.648009
     4.8990474   -3.6134505   -5.041441     1.8655863   -4.799377
    -5.074162     2.785183    -7.2079105    1.17119     -5.4221845
     3.4115486    1.2904737   -5.725713    -8.3982935    1.55972
     8.868936    -1.8804023   -1.188184     1.4222441    4.644693
    12.833991    -4.447345    -0.7109036   -0.9068291    3.0214996
     3.7415855 ]
  [  4.8351316    7.69565      0.53446746   5.8068285    5.2837653
    -1.5942345    5.1132054   -2.0102987    5.714198    -4.679373
     2.3857322    0.56943905   2.858938     8.861926     2.4583836
     0.39044502  -1.1101187    7.0288787    4.02846      5.0703363
     6.3214016   -6.6837225   -3.8799365    0.87895024  -6.911167
    -5.5402784   -0.31301427   0.88751     -5.59258     -2.5758967
    -4.57001     -5.4910097    6.344259     1.3278747   -0.8144084
    -5.8922853    6.2913237   -5.118787    -3.7647932   -8.911915
     2.7723482    0.20098352   1.0826368   -5.4491625    2.52587
    -6.4891376   -2.5728734    5.4348526   -2.388805    -7.7070055
     3.5134454   -8.221928     0.60799015   0.8563217    4.2456365
    -2.7917664   11.19659     -3.5343463    3.6952698   -0.13794172
     4.5141153   -5.301634    -4.500415   -10.654051    11.525325
    -6.7532673    3.0940204    3.0894082    3.9492526    1.7619723
     7.130692    -5.8603535   -3.542221     3.189839    -3.904769
    -5.2235384    2.8219967   -7.200142    -3.6182208   -3.4419484
     4.786948    -2.6611524   -3.2215147   -6.0934067   -3.021771
    10.337975    -2.7444239   -3.8255754   -0.8513925    6.167651
    12.783266     1.9053268   -3.2838812   -3.7842917    0.4220788
     4.4962306 ]
  [ 2.11124     9.483828   -4.0028214   7.8137255   1.3043883  -1.4620
  774
    3.5426664  -8.246752    5.0380054  -0.8392061   0.93319273  0.8279
  457
    6.835766    6.8444653   2.0859344   2.2857323   4.7759504   5.0065
  29
    3.5215988   2.5583365   2.7359743  -5.3954153   0.7415737  -4.1600
  385
   -2.6017106  -7.6315465   0.0951274  -0.6325534  -3.1517282  -5.4326
  324
   -2.2945213  -7.372366    4.316291    5.839383    7.4393015  -7.1489
  16
    4.9391837  -5.1775837  -0.8123046  -9.319872   -0.10712075  1.6155
  393
    0.9406865  -6.611548    0.2159088  -7.424259   -5.256575    3.8809
  543
   -5.324052   -7.554661    3.0403657  -9.510817    2.8584573  -2.0040
```

```
195
  6.022604   -1.424315   15.444103    0.20759457  7.0592575  -0.9541
354
  3.8307576  -5.1559825  -3.6347852  -7.97044    10.277409   -4.6591
043
  2.503798    1.6114237   3.2889302  -0.42354113  3.383179   -8.3468
55
 -6.2564354   7.2293243  -5.9056334  -2.4937172   0.40658253 -6.6710
362
 -2.3549647   1.6874005   1.4730852  -0.30552065 -4.403917   -6.0730
8
 -3.7269158  11.348831   -4.403538    0.07673979  1.231471    2.9534
67
 13.329096    3.338009   -3.6360824  -2.1201103   0.8428658   4.4126
37  ]
[ 3.4922473   4.4982095  -5.767888   -1.3178266   0.9186181  -2.3766
465
  0.7282289  -4.043273    5.633475    5.40027     4.1097255   2.6922
29
  7.580161    2.7529035   2.9928186   5.1495333  -0.3721658  -1.2233
295
 -0.30967396  2.5015922   8.4738655  -6.1870313   2.8067703   4.9828
043
 -4.226529   -5.076389    1.0651662  -6.193212   -0.01912975 -2.3474
698
 -3.2901037  -5.2008076  -0.47517014  0.3941641   4.3916306   0.0684
6589
  6.3354516  -7.1711116  -5.093693   -7.946332    1.615792    0.7823
3695
  3.3481863  -5.9747186  -2.400082   -7.007827   -8.431462   -1.3865
55
 -2.1267552  -2.0518672   0.7324561  -5.8410764   7.4181695   2.1141
23
  9.162542   -6.422852   14.786882    5.042012    7.0896044   2.9026
985
 -8.872308   -3.1997042  -3.7658138  -8.134214    5.7225847  -5.0084
853
  4.3039637   4.3468843   6.1780615  -1.1426599   5.953807   -8.7182
29
 -1.5459945   9.16362    -5.0024486  -0.6620538  -0.71198606 -6.4107
42
 -2.7763603   0.32479084  1.1726576  -2.5747085   0.88604873 -5.4698
8
 -3.6755822   9.3951845  -5.5535073  -5.11242     1.4563191   2.9579
09
  5.763336    5.259195   -2.4319663  -2.6575994   0.23322229  3.3525
33  ]
[ 2.734871    3.4061677  -2.7784667  -3.255712    0.12425268 -4.5458
646
 -1.2679465  -3.1154327   5.7746105   4.111931    4.763515    7.2436
54
  5.955747    3.2366219   1.321981    2.0538313  -3.6395123   7.6398
363
  0.09841228  3.9426343   2.5423596  -0.85184264  7.3786354   2.8326
418
 -3.8792372  -2.6616323  -5.3828716   5.7818604  -4.807608   -4.4328
513
 -9.434544   -7.3910246   1.5578136   4.838519    4.5788336  -4.8267
455
  1.6994786  -3.900304    3.0710964  -2.5125751   0.83175623 -0.5237
1645
```

```
    5.8800697   1.9861053  -1.3121216  -8.061008   -5.752239    2.6980
753
   -3.073857   -3.3162167   7.7250266  -7.420998   -1.3948529  12.9665
85
    4.270926   -0.6266554   9.059168    2.4238324   6.111846   -4.9994
864
   -5.1352544  -7.6435227  -9.975479   -9.462693    5.1849456  -2.1483
45
    5.049549    1.4663228   5.4030004   1.4082547   8.202511   -5.1619
87
   -5.3102417   2.0807996  -4.2401013  -2.5860047   1.0333769  -0.5106
1726
   -4.537824    5.6194277  -3.6344426   0.27441937  1.4413149   2.1943
104
   -3.0903225   7.4456377  -6.1604276  -7.0596933  -2.06486     2.6293
566
   -0.08701146  5.818083   -2.5382316  -1.4301207   1.4400984  -5.2362
275 ]
outros cavalos cavalos asininos muares vivos
[ 3.58808088e+00  8.90267754e+00  2.44048309e+00  3.95053768e+00
 -1.03410184e+00  4.81739521e+00  1.90770447e+00 -1.34829605e+00
  3.12164092e+00  8.50794792e+00  2.78923106e+00 -5.88034058e+00
  9.67856109e-01  1.20452499e+01 -3.69728088e+00  4.80988979e+00
 -2.45097828e+00 -1.91850984e+00  5.20131683e+00 -1.19534898e+00
  1.03487825e+00  6.13565540e+00 -2.42517591e+00  1.80349565e+00
  4.37107980e-01 -8.46675491e+00 -4.50074387e+00 -8.10868549e+00
  6.17140675e+00  2.62811637e+00  2.31927586e+00 -5.36333036e+00
  4.19952273e-02 -2.29659081e-02  1.22750032e+00 -6.24238586e+00
 -3.19425178e+00 -1.06047173e+01 -7.94883966e+00  1.63187146e+00
 -2.97473741e+00 -3.59007287e+00 -6.14373827e+00 -5.48138046e+00
 -7.72079849e+00 -1.58468771e+00  5.17795086e+00  1.23376122e+01
  4.76910114e+00  3.38924021e-01  5.90918350e+00 -1.23799477e+01
 -4.63227987e-01  9.36060429e-01  3.59902382e+00  2.31739926e+00
  7.87312984e-01 -5.59847832e+00  2.10112906e+00 -1.02241004e+00
  5.93431234e+00 -2.88797760e+00 -3.52675438e+00 -5.21362352e+00
 -1.73651147e+00 -5.42285872e+00 -3.07798386e-04 -4.36601448e+00
  9.17579055e-01  4.09947395e+00  2.36049557e+00  3.70721245e+00
 -6.33248758e+00 -1.06454544e+01  2.16816396e-01  1.94885409e+00
 -6.03120327e-01 -3.53743458e+00 -5.88157058e-01 -4.40862751e+00
  6.93835926e+00  9.47215843e+00 -3.82648301e+00 -5.85739326e+00
 -7.06617689e+00  9.83543587e+00  1.01875305e+01  1.68189466e+00
  6.62051821e+00  2.22021961e+00  9.86796379e+00 -4.60435104e+00
  2.03545356e+00  1.26429868e+00 -6.38830900e+00  4.05190182e+00]
[  6.7389107   13.2367115    4.0158653    0.09619975   5.5746617
   1.3567717    3.3321595   -8.300314     4.6122613   -2.879876
  -0.23158075  -1.179447     3.1384718   11.43513     -3.191731
   2.5970845   -2.8859587    4.5350647    2.0165915    4.536274
   3.9775789   -1.265662    -2.4021614    0.774514    -6.1615314
 -11.02191     -3.144893    -3.4779737   -4.911701     1.4669724
  -1.835757    -5.3360686    9.76886     -0.563184    -1.0922209
  -6.4391565    5.5208316   -5.49937     -4.6419964   -5.9516554
   4.936235     4.304922     1.6414716   -3.4948273   -2.2495565
  -6.019004    -5.944828     7.7668867   -1.0636795    0.6425177
   0.45978963  -7.275613    -0.7505648   -1.4642305    5.7461166
  -0.07269225   8.745312    -0.07751369  -0.4150784    1.7237929
  -2.4200659   -5.457046    -0.7292799   -7.8256044    5.3949685
  -4.0794497    0.12662238   1.3265862    0.22193754  -1.614884
   9.863603    -5.8877935   -1.8155496    4.2575583    2.951178
  -2.3952155    1.1418285   -7.061414    -4.8011937    2.7674804
   2.59795      5.9489684  -10.510402    -0.5993192    1.6428342
   6.6325793   -2.8875897   -2.8741329   -1.7434072    0.48758024
```

```
    15.923675    -5.217663    -4.532734    -6.6720347   -1.9660231
     9.657329  ]
[ 4.7512336   6.927713    1.943434    4.2231197   3.8263154  -0.8303
947
  0.9226888  -5.840002    5.082544   -4.956013    4.158126   -1.0683
227
  3.3354163   7.1475935  -2.6392453   3.130694    0.7107481   8.0168
61
  2.514673    3.2152264   0.47020292 -5.664296   -1.7775009   2.6923
857
 -5.6325617  -7.4422803  -0.39981103 -1.73315    -4.0786257  -1.8847
264
 -5.424461   -3.5527337   6.9930305   4.0763164   1.1567757  -4.5827
904
  5.1759644  -6.2927637  -4.5078793  -9.785501    0.8355565   2.9538
474
  0.30544114 -4.748681    3.1856663  -5.681303   -4.364521    6.8084
536
 -4.947859   -3.7375793   2.5160217  -7.739383    0.62479687 -0.8695
205
  7.8340855  -3.034377   11.970179   -1.6444497   2.2535505  -0.8174
211
  1.4890065  -3.2951272  -3.8488002  -8.439617   10.684834   -3.9758
568
  6.383353    4.1566725   3.304803    1.1789857   6.5016084  -5.5022
 -3.9088268   2.796331   -0.48442072 -4.887482    2.3393278  -7.3884
954
 -6.6391025   0.02128506  5.879231    0.17681897 -3.3269014  -7.2276
17
 -2.5736704  10.31285    -3.6517587  -2.240924    2.9645758   4.0383
058
 11.902599    2.0933323  -6.4502335  -6.640072   -0.5091186   3.7108
483 ]
[ 3.01321507e+00  8.96273136e+00 -3.37612653e+00  7.24109268e+00
  8.67813230e-01 -1.34828615e+00  2.31370425e+00 -8.79347420e+00
  5.54231977e+00 -1.01514030e+00  1.70598626e+00  1.08498752e-01
  7.01107121e+00  6.48974133e+00  1.13308573e+00  2.84205937e+00
  5.21314335e+00  5.23559761e+00  2.45946789e+00  2.32592821e+00
  1.16415548e+00 -4.67676020e+00  8.62288475e-01 -4.41780996e+00
 -2.03691149e+00 -8.47418594e+00  1.30894184e-02 -1.16064239e+00
 -3.35799074e+00 -5.15404415e+00 -1.88391864e+00 -6.74199581e+00
  4.07367182e+00  6.34464836e+00  7.53768873e+00 -6.30867863e+00
  5.12233734e+00 -5.44381618e+00 -1.30389261e+00 -9.43997002e+00
  7.84013271e-02  1.99358869e+00  1.39033806e+00 -6.17307663e+00
  6.45139515e-02 -6.96581459e+00 -5.52944326e+00  3.83234096e+00
 -6.15654993e+00 -6.29358006e+00  3.23453164e+00 -9.18755531e+00
  3.10082579e+00 -2.03874063e+00  6.41712904e+00 -1.09025693e+00
  1.56571331e+01  5.94869673e-01  7.16469765e+00 -9.38162565e-01
  3.63253117e+00 -4.40825653e+00 -4.22993612e+00 -7.91588211e+00
  1.03815289e+01 -4.06584358e+00  3.17907882e+00  1.46107817e+00
  3.34142423e+00  6.06352389e-02  2.72377586e+00 -8.30867004e+00
 -6.17044687e+00  7.14053345e+00 -4.55290842e+00 -3.33679342e+00
  6.67264700e-01 -6.40364170e+00 -2.83874774e+00  1.91225171e+00
  1.68763292e+00  2.71907806e-01 -5.00337601e+00 -6.43370295e+00
 -3.83162999e+00  1.11854191e+01 -4.58447313e+00  2.53238261e-01
  2.21270275e+00  2.33332348e+00  1.34380531e+01  3.01664066e+00
 -4.65133429e+00 -3.78400826e+00  1.04440674e-01  4.36304474e+00]
[ 3.5769923   4.5197716  -5.7257557  -1.3722883   0.9032966  -2.3743
832
  0.7297734  -4.0348005   5.6722536   5.3275156   4.175699    2.6428
838
```

```
  7.582141    2.763667    2.8945103   5.151905   -0.3315044  -1.1753
385
 -0.37650722  2.5296836   8.321567   -6.1195226   2.7669      4.9479
78
 -4.1922197  -5.222511    1.0607603  -6.219198   -0.05478477 -2.3793
044
 -3.250962   -5.149919   -0.5364835   0.4198482   4.46568     0.0718
3284
  6.3715243  -7.183168   -5.099803   -8.001589    1.6404064   0.8477
406
  3.3878882  -6.0278034  -2.318263   -7.0168552  -8.403177   -1.4195
18
 -2.1642952  -1.9849641   0.7243173  -5.8889174   7.4004707   2.1493
66
  9.175852   -6.358588   14.840786    5.049617    7.105885    2.9161
198
 -8.82369    -3.1334066  -3.8323774  -8.157346    5.7064867  -4.9427
28
  4.3598595   4.3366814   6.1586847  -1.0653625   5.888731   -8.7095
65
 -1.5298921   9.149619   -4.9647517  -0.7091187  -0.68653214 -6.4037
466
 -2.8800862   0.30682647  1.191835   -2.4818141   0.8957044  -5.5368
84
 -3.6737857   9.431527   -5.594685   -5.0576363   1.5047975   2.8909
86
  5.7357225   5.2505064  -2.5311267  -2.8046482   0.26745036  3.3451
16  ]
[ 2.734871    3.4061677  -2.7784667  -3.255712    0.12425268 -4.5458
646
 -1.2679465  -3.1154327   5.7746105   4.111931    4.763515    7.2436
54
  5.955747    3.2366219   1.321981    2.0538313  -3.6395123   7.6398
363
  0.09841228  3.9426343   2.5423596  -0.85184264  7.3786354   2.8326
418
 -3.8792372  -2.6616323  -5.3828716   5.7818604  -4.807608   -4.4328
513
 -9.434544   -7.3910246   1.5578136   4.838519    4.5788336  -4.8267
455
  1.6994786  -3.900304    3.0710964  -2.5125751   0.83175623 -0.5237
1645
  5.8800697   1.9861053  -1.3121216  -8.061008   -5.752239    2.6980
753
 -3.073857   -3.3162167   7.7250266  -7.420998   -1.3948529  12.9665
85
  4.270926   -0.6266554   9.059168    2.4238324   6.111846   -4.9994
864
 -5.1352544  -7.6435227  -9.975479   -9.462693    5.1849456  -2.1483
45
  5.049549    1.4663228   5.4030004   1.4082547   8.202511   -5.1619
87
 -5.3102417   2.0807996  -4.2401013  -2.5860047   1.0333769  -0.5106
1726
 -4.537824    5.6194277  -3.6344426   0.27441937  1.4413149   2.1943
104
 -3.0903225   7.4456377  -6.1604276  -7.0596933  -2.06486     2.6293
566
 -0.08701146  5.818083   -2.5382316  -1.4301207   1.4400984  -5.2362
275 ]
```

In [42]:

```
embeddings = [nlp(doc).vector for doc in corpus]
```

In [43]:

```
len(embeddings)
```

Out[43]:

```
10147
```

In [44]:

```
embeddings[0].shape
```

Out[44]:

```
(96,)
```

In [45]:

```
embeddings[10].shape
```

Out[45]:

```
(96,)
```

In [46]:

```
from sklearn.preprocessing import LabelBinarizer

encoder = LabelBinarizer()

labels = encoder.fit_transform(y)
```

In [47]:

```
encoder.classes_
```

Out[47]:

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 1
6, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 3
3, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 5
0, 51,
       52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 6
7, 68,
       69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 80, 81, 82, 83, 84, 8
5, 86,
       87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97])
```

In [48]:

```
labels[0]
```

Out[48]:

```
array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
       0, 0, 0, 0, 0, 0, 0, 0])
```

In [49]:

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Dropout
model = Sequential()
model.add(Dense(512, activation='relu', kernel_initializer='he_normal', input_di
m=96))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu', kernel_initializer='he_normal'))
model.add(Dropout(0.2))
model.add(Dense(len(encoder.classes_), activation='softmax'))
```

In [50]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 512)               49664
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 96)                49248
=================================================================
Total params: 361,568
Trainable params: 361,568
Non-trainable params: 0
_________________________________________________________________
```

In [51]:

```
model.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['acc'])
```

In [52]:

```
model.fit(np.array(embeddings), labels, epochs=20)
```

```
Train on 10147 samples
Epoch 1/20
10147/10147 [==============================] - 3s 306us/sample - los
s: 3.4808 - acc: 0.3062
Epoch 2/20
10147/10147 [==============================] - 0s 46us/sample - los
s: 2.4107 - acc: 0.4625
Epoch 3/20
10147/10147 [==============================] - 0s 48us/sample - los
s: 1.9680 - acc: 0.5446
Epoch 4/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 1.6807 - acc: 0.6025
Epoch 5/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 1.4850 - acc: 0.6448
Epoch 6/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 1.3324 - acc: 0.6775
Epoch 7/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 1.1849 - acc: 0.7120
Epoch 8/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 1.0823 - acc: 0.7314
Epoch 9/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.9972 - acc: 0.7509
Epoch 10/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.9110 - acc: 0.7746
Epoch 11/20
10147/10147 [==============================] - 0s 46us/sample - los
s: 0.8461 - acc: 0.7805
Epoch 12/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.7755 - acc: 0.8035
Epoch 13/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.7356 - acc: 0.8074
Epoch 14/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.6831 - acc: 0.8234
Epoch 15/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.6396 - acc: 0.8374
Epoch 16/20
10147/10147 [==============================] - 0s 46us/sample - los
s: 0.5977 - acc: 0.8449
Epoch 17/20
10147/10147 [==============================] - 0s 46us/sample - los
s: 0.5680 - acc: 0.8551
Epoch 18/20
10147/10147 [==============================] - 0s 46us/sample - los
s: 0.5253 - acc: 0.8641
Epoch 19/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.5041 - acc: 0.8694
Epoch 20/20
10147/10147 [==============================] - 0s 47us/sample - los
s: 0.4886 - acc: 0.8728
```

Out[52]:

```
<tensorflow.python.keras.callbacks.History at 0x7f0fe00b37f0>
```

In [53]:

```
y_pred = model.predict(np.array([nlp('arruelas plastico').vector]))
```

In [54]:

```
y_pred.argmax()
```

Out[54]:

```
83
```

In [55]:

```
y_pred = model.predict(np.array([nlp('bolsas plastico').vector]))
```

In [56]:

```
y_pred.argmax()
```

Out[56]:

```
78
```

In [57]:

```
y_pred = model.predict(np.array([nlp('bolsas couro').vector]))
```

In [58]:

```
y_pred.argmax()
```

Out[58]:

```
17
```

In [59]:

```
embeddings_complete = []
for doc in corpus:
    tokens = nlp(doc)
    doc_embedding = []
    for token in tokens:
        if token.has_vector:
            doc_embedding.append(token.vector)
    embeddings_complete.append(doc_embedding)
```

In [60]:

```
max_len = 0
min_len = 1000
avg_len = 0
for emb in embeddings_complete:
    size = len(emb)
    avg_len += size
    if size > max_len:
        max_len = size
    if size < min_len:
        min_len = size

avg_len = avg_len / len(embeddings_complete)
```

In [61]:

```
min_len
```

Out[61]:

```
1
```

In [62]:

```
avg_len
```

Out[62]:

```
23.768897210998325
```

In [63]:

```
max_len
```

Out[63]:

```
98
```

In [64]:

```
print(len(embeddings_complete))
```

```
10147
```

In [65]:

```
print(embeddings_complete[0])
```

```
[array([ 6.64293242e+00,  1.87291443e-01,  6.24938774e+00,  1.258316
28e+00,
        1.12645035e+01,  1.62478197e+00, -6.70816994e+00, -1.2303943
6e+00,
        5.05357218e+00, -3.12805176e+00,  1.66952193e+00, -3.9707720
3e+00,
        4.87114954e+00,  7.86261654e+00, -6.47044659e+00,  1.9473476
4e+00,
        2.06741238e+00,  6.61689878e-01,  3.94642997e+00, -1.6330286
9e-02,
        6.39751863e+00, -3.85731220e-01, -5.53474426e-01,  3.1952168
9e+00,
        1.31331623e-01, -1.11653461e+01, -1.57218754e+00, -5.5233508
3e-01,
        1.86404061e+00, -2.52212238e+00, -4.66619825e+00, -7.2374286
7e+00,
        8.59963608e+00,  1.08069718e+00,  6.40867996e+00, -8.2045192
7e+00,
        5.25842619e+00, -9.63336372e+00, -2.79402876e+00,  1.0287023
8e+00,
        1.69568944e+00,  4.14215183e+00, -3.32162738e+00,  4.0781986
7e-01,
       -6.81309938e+00, -9.28379536e+00, -2.13685131e+00,  7.5763745
3e+00,
        4.28763151e+00,  3.81842184e+00, -5.71094632e-01, -1.0639370
9e+01,
       -8.56875801e+00, -5.78028321e-01,  5.24746656e+00,  3.4625349
0e+00,
        1.76879239e+00, -4.92994595e+00, -2.56430328e-01, -5.3654444
2e-01,
       -6.48282909e+00, -1.30999970e+00, -8.14078450e-01, -6.3433513
6e+00,
        4.05887890e+00, -3.61303830e+00,  1.39137745e+00,  2.3055462
8e+00,
        1.39408684e+00, -2.85495901e+00,  1.12897825e+01, -1.9838700
3e+00,
        2.19006634e+00, -3.19921970e-03,  4.93487692e+00, -2.7595713
1e+00,
        5.29434919e+00, -4.04155397e+00, -2.30014086e+00,  1.6897634
3e+00,
       -1.23612642e-01,  4.63714743e+00, -3.70339537e+00,  2.0604894
2e+00,
       -6.55541229e+00,  4.76594925e+00, -6.29005575e+00,  4.8809340
6e-01,
        7.20721579e+00, -2.39367485e+00,  1.37540264e+01, -6.4406385
4e+00,
       -7.21631908e+00, -3.75078368e+00,  5.73268771e-01,  3.1040749
5e+00],
      dtype=float32), array([  1.0863048 ,  -2.7894533 ,   6.9680557
,   0.08295405,
         8.480454  ,  -0.23761934,  -0.68354964,  -5.7243595 ,
        -5.0674005 ,   4.1179113 ,   2.51198   ,  -5.6980796 ,
        -0.24173518,  -3.6367745 ,   4.9928293 ,   6.2197914 ,
         2.15706   ,  -4.435322  ,   4.0617003 ,   0.9087716 ,
         4.7484436 ,   2.668272  ,   6.628061  ,   7.500891  ,
        -3.9301577 ,  -8.1157875 ,   3.061476  ,   1.6057082 ,
         0.2058376 ,  -4.239825  ,  -7.7303863 ,  -4.3014393 ,
         7.102921  ,   4.908053  ,   6.276469  ,  -3.8917956 ,
         5.7583046 ,  -1.0862677 ,   6.300334  ,   0.75612783,
        -4.88805   ,  -0.766531  ,  -3.4806933 ,   3.9001112 ,
        -5.8072376 ,  -0.11451912,  -1.4290804 ,   6.4062853 ,
```

```
          -6.078262  ,  -2.2355113 ,  -3.9690788 ,  -6.9767547 ,
           2.033164  ,   3.0988002 ,   2.6800663 ,   4.3491077 ,
          -0.18103781,   4.9755144 ,   1.0837207 ,  -6.5856857 ,
          -6.095976  ,   0.3742739 ,   3.2784503 ,  -3.587216  ,
          11.426226  ,  -2.8507655 ,  -1.9860098 ,   4.735774  ,
           1.7975597 ,  -0.62502664,  -4.2821817 ,  -4.464139  ,
          -4.9245424 ,  -8.896518  ,   0.8314903 ,  -0.251832  ,
           6.267582  ,  -0.5932001 ,   2.478343  ,   2.056869  ,
           7.0361414 ,  -6.071352  ,  -4.1507573 ,   3.7893767 ,
          -1.1372217 ,   3.4948878 ,  -6.7170315 ,  -3.7457006 ,
           3.6472187 ,   9.098037  ,   0.47309887,  -3.617497  ,
         -12.048483  ,   0.0188278 ,   3.887532  ,  -6.6303854 ],
        dtype=float32), array([  2.1992743 ,   8.091729  ,   4.522311
  ,  -8.188901  ,
          -7.0303016 ,   0.45658055,   4.826764  ,  -3.585285  ,
          -7.113357  ,  -3.4468498 ,  -2.8221853 ,  -3.3022718 ,
           1.9646587 ,  -2.1140697 ,  -9.23862   ,   6.680383  ,
          -4.5580435 ,   2.3379543 ,   4.260918  ,   5.518238  ,
           5.128637  ,   3.8368454 ,   6.381139  ,   5.4880304 ,
         -12.824165  ,   1.170821  ,   3.2437282 ,  -2.5353253 ,
          -0.12281916,   3.536547  ,   3.3140566 ,  -7.2784824 ,
           1.7457491 ,  -0.7575569 ,   0.47740614,   4.1811776 ,
           5.8105583 ,   1.6979222 ,  -3.199921  ,  -7.532384  ,
          -3.2352204 ,  -1.7950699 ,  -3.8054972 ,  -2.0748372 ,
          -0.7112931 ,   7.410285  ,  -3.1889167 ,  -7.5247235 ,
          -0.6689456 ,  -4.497854  ,  -0.52249455,  -6.6036496 ,
           9.145586  ,  -0.80700564,   1.5849553 ,  -2.0926564 ,
          11.816529  ,   4.875844  ,   3.7209067 ,   3.7519534 ,
           1.0406731 ,  -2.5321631 ,   4.894824  ,  -0.89243233,
          -0.5185905 ,  -7.3550806 ,   9.400541  ,   3.9300883 ,
          -0.54305255,  -5.671421  ,   1.1497175 ,  -2.7825446 ,
          -5.751813  ,  -2.8366346 ,  -3.3904161 ,   2.5586598 ,
           0.456926  ,  -3.3449183 ,   6.023432  ,  -0.99129224,
           4.262294  ,  -4.4304085 ,  -2.8050134 ,  -2.4955997 ,
          -1.8132106 ,   4.870413  ,  -1.4493029 ,   1.2932534 ,
           9.012482  ,   5.160825  ,   0.860921  ,  -7.857154  ,
           1.8112178 ,   2.6701531 ,   1.7830403 ,  -4.2964616 ],
        dtype=float32), array([ -0.73945105,   2.8385053 ,   1.0226985
  ,  -0.8699119 ,
           0.52795815,  -1.3864001 ,   4.200994  ,  -0.83615595,
           6.2351303 ,  -5.654449  ,  -5.499399  ,  -0.74021846,
           8.0443325 ,  10.607567  ,   0.6549323 ,  -0.64773285,
          -3.5004978 ,   7.270165  ,   2.1535282 ,   2.1119525 ,
           7.592078  ,  -3.0291953 ,  -4.9867024 ,   6.358389  ,
          -5.504796  ,   0.8302175 ,  -2.2097998 ,  -0.89226305,
          -5.209181  ,   2.8009102 ,  -0.6648627 ,  -5.170225  ,
           2.3313642 ,  -1.2005141 ,  -0.17824328, -10.282638  ,
           5.1335278 ,  -4.600122  ,  -5.960139  ,  -7.4703474 ,
           2.422846  ,   0.66232747,   1.0053172 ,  -0.69547033,
          -2.6582355 ,  -5.127459  ,  -5.0771008 ,   0.898503  ,
           1.7458392 ,  -4.2765102 ,   0.84111696,  -4.7694664 ,
           4.3065453 ,   0.45421484,   5.332529  ,  -0.11407435,
          10.148493  ,  -3.6055446 ,   3.6794648 ,  -1.1967767 ,
           8.061204  ,  -4.457557  ,  -1.1100774 ,  -9.182857  ,
          10.561043  , -10.291653  ,   7.2363005 ,   0.9260359 ,
           2.4826021 ,   5.648009  ,   4.8990474 ,  -3.6134505 ,
          -5.041441  ,   1.8655863 ,  -4.799377  ,  -5.074162  ,
           2.785183  ,  -7.2079105 ,   1.17119   ,  -5.4221845 ,
           3.4115486 ,   1.2904737 ,  -5.725713  ,  -8.3982935 ,
           1.55972   ,   8.868936  ,  -1.8804023 ,  -1.188184  ,
           1.4222441 ,   4.644693  ,  12.833991  ,  -4.447345  ,
```

```
        -0.7109036 ,  -0.9068291 ,   3.0214996 ,   3.7415855 ],
      dtype=float32), array([  4.8351316 ,   7.69565   ,   0.5344674
6,   5.8068285 ,
         5.2837653 ,  -1.5942345 ,   5.1132054 ,  -2.0102987 ,
         5.714198  ,  -4.679373  ,   2.3857322 ,   0.56943905,
         2.858938  ,   8.861926  ,   2.4583836 ,   0.39044502,
        -1.1101187 ,   7.0288787 ,   4.02846   ,   5.0703363 ,
         6.3214016 ,  -6.6837225 ,  -3.8799365 ,   0.87895024,
        -6.911167  ,  -5.5402784 ,  -0.31301427,   0.88751   ,
        -5.59258   ,  -2.5758967 ,  -4.57001   ,  -5.4910097 ,
         6.344259  ,   1.3278747 ,  -0.8144084 ,  -5.8922853 ,
         6.2913237 ,  -5.118787  ,  -3.7647932 ,  -8.911915  ,
         2.7723482 ,   0.20098352,   1.0826368 ,  -5.4491625 ,
         2.52587   ,  -6.4891376 ,  -2.5728734 ,   5.4348526 ,
        -2.388805  ,  -7.7070055 ,   3.5134454 ,  -8.221928  ,
         0.60799015,   0.8563217 ,   4.2456365 ,  -2.7917664 ,
        11.19659   ,  -3.5343463 ,   3.6952698 ,  -0.13794172,
         4.5141153 ,  -5.301634  ,  -4.500415  , -10.654051  ,
        11.525325  ,  -6.7532673 ,   3.0940204 ,   3.0894082 ,
         3.9492526 ,   1.7619723 ,   7.130692  ,  -5.8603535 ,
        -3.542221  ,   3.189839  ,  -3.904769  ,  -5.2235384 ,
         2.8219967 ,  -7.200142  ,  -3.6182208 ,  -3.4419484 ,
         4.786948  ,  -2.6611524 ,  -3.2215147 ,  -6.0934067 ,
        -3.021771  ,  10.337975  ,  -2.7444239 ,  -3.8255754 ,
        -0.8513925 ,   6.167651  ,  12.783266  ,   1.9053268 ,
        -3.2838812 ,  -3.7842917 ,   0.4220788 ,   4.4962306 ],
      dtype=float32), array([ 2.11124   ,  9.483828  , -4.0028214 ,
7.8137255 ,  1.3043883 ,
       -1.4620774 ,  3.5426664 , -8.246752  ,  5.0380054 , -0.839206
1 ,
        0.93319273,  0.8279457 ,  6.835766  ,  6.8444653 ,  2.085934
4 ,
        2.2857323 ,  4.7759504 ,  5.006529  ,  3.5215988 ,  2.558336
5 ,
        2.7359743 , -5.3954153 ,  0.7415737 , -4.1600385 , -2.601710
6 ,
       -7.6315465 ,  0.0951274 , -0.6325534 , -3.1517282 , -5.432632
4 ,
       -2.2945213 , -7.372366  ,  4.316291  ,  5.839383  ,  7.439301
5 ,
       -7.148916  ,  4.9391837 , -5.1775837 , -0.8123046 , -9.319872
,
       -0.10712075,  1.6155393 ,  0.9406865 , -6.611548  ,  0.215908
8 ,
       -7.424259  , -5.256575  ,  3.8809543 , -5.324052  , -7.554661
,
        3.0403657 , -9.510817  ,  2.8584573 , -2.0040195 ,  6.022604
,
       -1.424315  , 15.444103  ,  0.20759457,  7.0592575 , -0.954135
4 ,
        3.8307576 , -5.1559825 , -3.6347852 , -7.97044   , 10.277409
,
       -4.6591043 ,  2.503798  ,  1.6114237 ,  3.2889302 , -0.423541
13,
        3.383179  , -8.346855  , -6.2564354 ,  7.2293243 , -5.905633
4 ,
       -2.4937172 ,  0.40658253, -6.6710362 , -2.3549647 ,  1.687400
5 ,
        1.4730852 , -0.30552065, -4.403917  , -6.07308   , -3.726915
8 ,
       11.348831  , -4.403538  ,  0.07673979,  1.231471  ,  2.953467
```

```
,
       13.329096  ,  3.338009  , -3.6360824 , -2.1201103 ,  0.842865
8 ,
        4.412637  ], dtype=float32), array([ 3.4922473 ,  4.4982095
, -5.767888  , -1.3178266 ,  0.9186181 ,
       -2.3766465 ,  0.7282289 , -4.043273  ,  5.633475  ,  5.40027
,
        4.1097255 ,  2.692229  ,  7.580161  ,  2.7529035 ,  2.992818
6 ,
        5.1495333 , -0.3721658 , -1.2233295 , -0.30967396,  2.501592
2 ,
        8.4738655 , -6.1870313 ,  2.8067703 ,  4.9828043 , -4.226529
,
       -5.076389  ,  1.0651662 , -6.193212  , -0.01912975, -2.347469
8 ,
       -3.2901037 , -5.2008076 , -0.47517014,  0.3941641 ,  4.391630
6 ,
        0.06846589,  6.3354516 , -7.1711116 , -5.093693  , -7.946332
,
        1.615792  ,  0.78233695,  3.3481863 , -5.9747186 , -2.400082
,
       -7.007827  , -8.431462  , -1.386555  , -2.1267552 , -2.051867
2 ,
        0.7324561 , -5.8410764 ,  7.4181695 ,  2.114123  ,  9.162542
,
       -6.422852  , 14.786882  ,  5.042012  ,  7.0896044 ,  2.902698
5 ,
       -8.872308  , -3.1997042 , -3.7658138 , -8.134214  ,  5.722584
7 ,
       -5.0084853 ,  4.3039637 ,  4.3468843 ,  6.1780615 , -1.142659
9 ,
        5.953807  , -8.718229  , -1.5459945 ,  9.16362   , -5.002448
6 ,
       -0.6620538 , -0.71198606, -6.410742  , -2.7763603 ,  0.324790
84,
        1.1726576 , -2.5747085 ,  0.88604873, -5.46988   , -3.675582
2 ,
        9.3951845 , -5.5535073 , -5.11242   ,  1.4563191 ,  2.957909
,
        5.763336  ,  5.259195  , -2.4319663 , -2.6575994 ,  0.233222
29,
        3.352533  ], dtype=float32), array([ 2.734871  ,  3.4061677
, -2.7784667 , -3.255712  ,  0.12425268,
       -4.5458646 , -1.2679465 , -3.1154327 ,  5.7746105 ,  4.111931
,
        4.763515  ,  7.243654  ,  5.955747  ,  3.2366219 ,  1.321981
,
        2.0538313 , -3.6395123 ,  7.6398363 ,  0.09841228,  3.942634
3 ,
        2.5423596 , -0.85184264,  7.3786354 ,  2.8326418 , -3.879237
2 ,
       -2.6616323 , -5.3828716 ,  5.7818604 , -4.807608  , -4.432851
3 ,
       -9.434544  , -7.3910246 ,  1.5578136 ,  4.838519  ,  4.578833
6 ,
       -4.8267455 ,  1.6994786 , -3.900304  ,  3.0710964 , -2.512575
1 ,
        0.83175623, -0.52371645,  5.8800697 ,  1.9861053 , -1.312121
6 ,
       -8.061008  , -5.752239  ,  2.6980753 , -3.073857  , -3.316216
7 ,
```

```
        7.7250266 , -7.420998  , -1.3948529 , 12.966585  ,  4.270926
,
       -0.6266554 ,  9.059168  ,  2.4238324 ,  6.111846  , -4.999486
4 ,
       -5.1352544 , -7.6435227 , -9.975479  , -9.462693  ,  5.184945
6 ,
       -2.148345  ,  5.049549  ,  1.4663228 ,  5.4030004 ,  1.408254
7 ,
        8.202511  , -5.161987  , -5.3102417 ,  2.0807996 , -4.240101
3 ,
       -2.5860047 ,  1.0333769 , -0.51061726, -4.537824  ,  5.619427
7 ,
       -3.6344426 ,  0.27441937,  1.4413149 ,  2.1943104 , -3.090322
5 ,
        7.4456377 , -6.1604276 , -7.0596933 , -2.06486   ,  2.629356
6 ,
       -0.08701146,  5.818083  , -2.5382316 , -1.4301207 ,  1.440098
4 ,
       -5.2362275 ], dtype=float32)]
```

In [1]:

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv1D, Dense, MaxPooling1D, Flatten, Input
model = Sequential()
model.add(Input(shape=(50, 96)))
model.add(Conv1D(128, 10, activation='relu'))
model.add(MaxPooling1D(5))
model.add(Conv1D(256, 2, activation='relu'))
model.add(MaxPooling1D(7))
model.add(Flatten())
model.add(Dense(96, activation='softmax'))

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 41, 128)           123008
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 8, 128)            0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 7, 256)            65792
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 1, 256)            0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 96)                24672
=================================================================
Total params: 213,472
Trainable params: 213,472
Non-trainable params: 0
_________________________________________________________________
```

In [2]:

```
model.compile(loss='categorical_crossentropy',
              optimizer='rmsprop',
              metrics=['acc'])
```

In [3]:

```
Xemb = np.zeros((len(embeddings_complete), 50, 96))
for ind, emb in enumerate(embeddings_complete):
    embarray = np.array(emb[:50])
    Xemb[ind, :embarray.shape[0], :] = embarray
```

```
---------------------------------------------------------------------
-------
NameError                                 Traceback (most recent cal
l last)
<ipython-input-3-b8131d7d996a> in <module>
----> 1 Xemb = np.zeros((len(embeddings_complete), 50, 96))
      2 for ind, emb in enumerate(embeddings_complete):
      3     embarray = np.array(emb[:50])
      4     Xemb[ind, :embarray.shape[0], :] = embarray

NameError: name 'np' is not defined
```

In [69]:

```
#with open('xemb.npy', 'bw') as xout:
#    np.save(xout, Xemb)
```

In [70]:

```
#with open('labels.npy', 'bw') as yout:
#    np.save(yout, labels)
```

In [3]:

```
import numpy as np
Xemb = np.load('xemb.npy', allow_pickle=True)
```

In [4]:

```
labels = np.load('labels.npy')
```

In [5]:

```
from sklearn.model_selection import train_test_split
x_train, x_val, y_train, y_val = train_test_split(Xemb, labels)
```

In [28]:

```
model.fit(x_train, y_train, validation_data=(x_val, y_val),
          epochs=5, batch_size=128)
```

```
Train on 7610 samples, validate on 2537 samples
Epoch 1/5
7610/7610 [==============================] - 1s 88us/sample - loss:
0.1224 - acc: 0.9867 - val_loss: 0.2361 - val_acc: 0.9519
Epoch 2/5
7610/7610 [==============================] - 1s 83us/sample - loss:
0.0738 - acc: 0.9867 - val_loss: 0.2886 - val_acc: 0.9401
Epoch 3/5
7610/7610 [==============================] - 1s 82us/sample - loss:
0.0709 - acc: 0.9921 - val_loss: 0.2414 - val_acc: 0.9566
Epoch 4/5
7610/7610 [==============================] - 1s 82us/sample - loss:
0.0733 - acc: 0.9894 - val_loss: 0.2326 - val_acc: 0.9527
Epoch 5/5
7610/7610 [==============================] - 1s 83us/sample - loss:
0.0260 - acc: 0.9949 - val_loss: 0.2617 - val_acc: 0.9492
```

Out[28]:

```
<tensorflow.python.keras.callbacks.History at 0x7efd2bdc9470>
```

In [29]:

```
search = np.zeros((1, 50, 96))
search[0, 0, :] = np.array(nlp('arruelas').vector)
search[0, 1, :] = np.array(nlp('plastico').vector)

y_pred = model.predict(search)
y_pred.argmax()
```

Out[29]:

```
24
```

In [34]:

```
search = np.zeros((1, 50, 96))
search[0, 0, :] = np.array(nlp('brinquedos').vector)
search[0, 1, :] = np.array(nlp('bonecos').vector)
search[0, 2, :] = np.array(nlp('miniaturas').vector)

y_pred = model.predict(search)
y_pred.argmax()
```

Out[34]:

```
25
```

In [35]:

```
search
```

Out[35]:

```
array([[[11.12016296,  6.8395319 , -1.83967328, ..., -8.09624195,
         -0.34083176,  2.46474552],
        [13.89891243,  2.83249092, -3.4532814 , ..., -5.92721653,
         -1.70419478,  3.98256636],
        [14.92188644, -5.05593729, -1.24579597, ..., -6.66526794,
          0.09027869,  0.22857964],
        ...,
        [ 0.        ,  0.        ,  0.        , ...,  0.        ,
          0.        ,  0.        ],
        [ 0.        ,  0.        ,  0.        , ...,  0.        ,
          0.        ,  0.        ],
        [ 0.        ,  0.        ,  0.        , ...,  0.        ,
          0.        ,  0.        ]]])
```

In [ ]:

```

```

In [ ]:

```

```