



Computadores

- **São capazes apenas de processar números**
- **Não lidam muito bem com informação não estruturada!**



Computadores

- É preciso criar formas de representação de texto as quais os computadores sejam capazes de processar!
 - Transformar texto em uma representação numérica
 - Transformar informação não estruturada em estruturada

```
,
    this.cycle(true)
}

this.interval = clearInterval(
    return this
}

Carousel.prototype.next = function() {
    if (this.sliding) return
    return this.slide('next')
}

Carousel.prototype.prev = function() {
    if (this.sliding) return
    return this.slide('prev')
}

Carousel.prototype.slide = function(
    var $active = this.$element.
    var $next = next || this.g
    var isCycling = this.interval
    var direction = type == 'next'
    var fallback = type == 'next'
    var that = this

    if (!$next.length) {
        if (!this.options.wrap) return
        $next = this.$element.find('
    }

    if ($next.hasClass('active'))

    var relatedTarget = $next[0]
    var slideEvent = $.Event('slide
        relatedTarget: relatedTarget,
        direction: direction
    ))
    this.$element.trigger(slideEvent
```

```
$sort_order = array();

foreach ($quotes as $key => $value) {
    $sort_order[$key] = $value['sort_order'];
}

array_multisort($sort_order, SORT_ASC, $quotes);

$this->session->data['lpa']['shipping_methods'] = $quotes;
$this->session->data['lpa']['address'] = $address;

if (empty($quotes)) {
    $json['error'] = $this->language->get('
        error_no_shipping_methods');
} else {
    $json['quotes'] = $quotes;
}

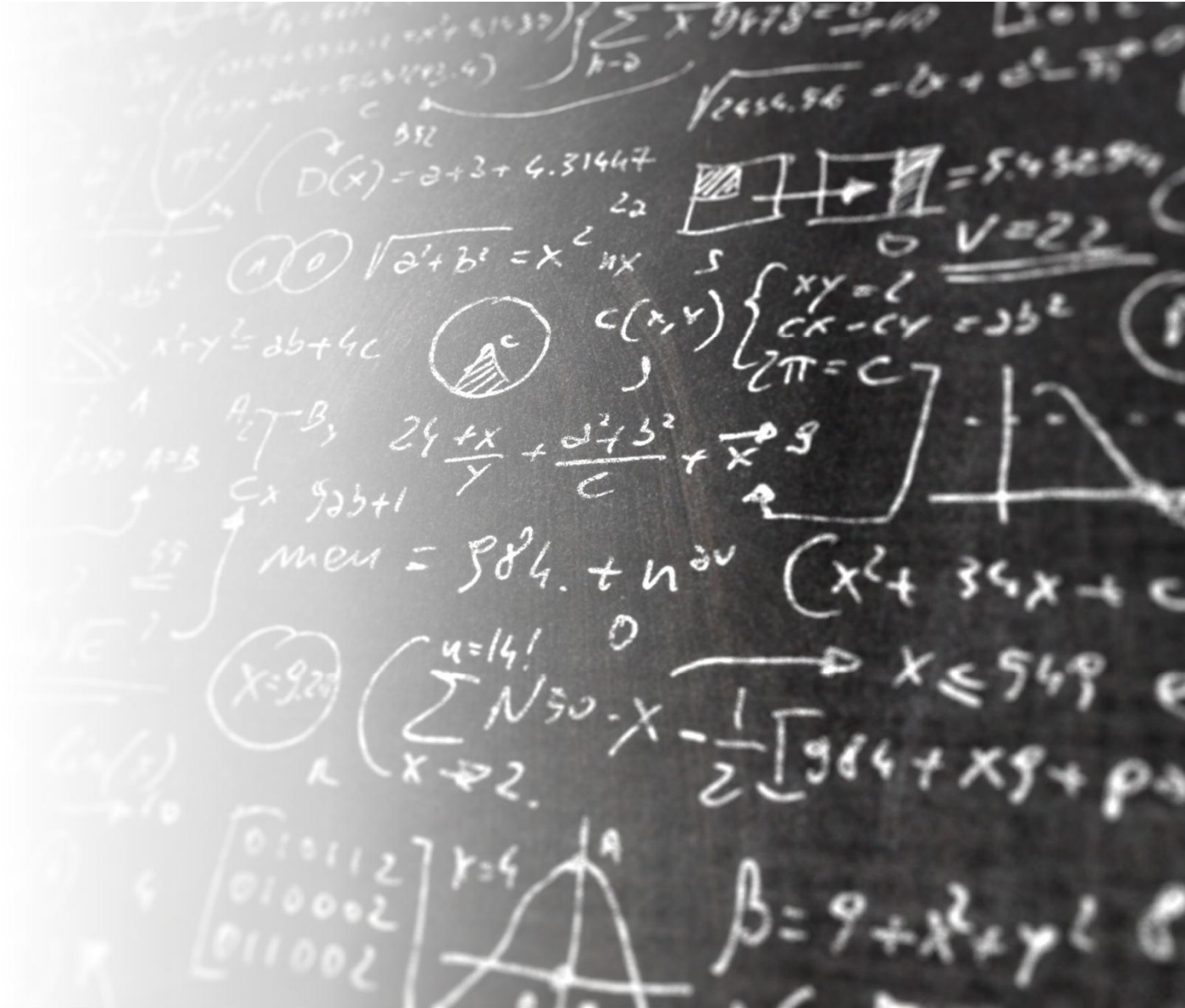
if (isset($this->session->data['lpa']['shipping_method']) && !
    empty($this->session->data['lpa']['shipping_method']) &&
    isset($this->session->data['lpa']['shipping_method']['code']
    )) {
    $json['selected'] = $this->session->data['lpa']['
        shipping_method']['code'];
} else {
    $json['selected'] = '';
}

} else {
    $json['error'] = $this->language->get('error_shipping_methods');
}

$this->response->addHeader('Content-Type: application/json');
```

Word Embedding


- Representações computacionais de texto
 - Independentes de contexto
 - Dependentes de contexto



One Hot Encoding

- Nossa vida é controlada por algoritmos

	algoritmos	controlada	é	Nossa	por	vida
Nossa	0	0	0	1	0	0
vida	0	0	0	0	0	1
é	0	0	1	0	0	0
controlada	0	1	0	0	0	0
por	0	0	0	0	1	0
algoritmos	1	0	0	0	0	0



TF-IDF

- Processos Judiciais
 - Comum: Processo, lei, partes
 - Incomum: Ambiental



Qual processo é da área ambiental?

- Doc A: ambiental 0x
- Doc B: ambiental 6x

	TF	IDF	TF-IDF (*)
A	$0/7=0$	$\text{Log}(2/1)=0.69$	0
B	$6/7=0,85$		0.5865

['ambiental' 'de' 'jurídico' 'lei' 'processo' 'segundo' 'trabalhista']

(0, 3) 0.26469629755874713

(0, 5) 0.4146978997095072

(0, 0) 0.8293957994190144

(0, 4) 0.26469629755874713

(1, 2) 0.7772211620785797

(1, 4) 0.6292275146695526

(2, 1) 0.6406554311067799

(2, 2) 0.5051001005334584

(2, 3) 0.4089220628888078

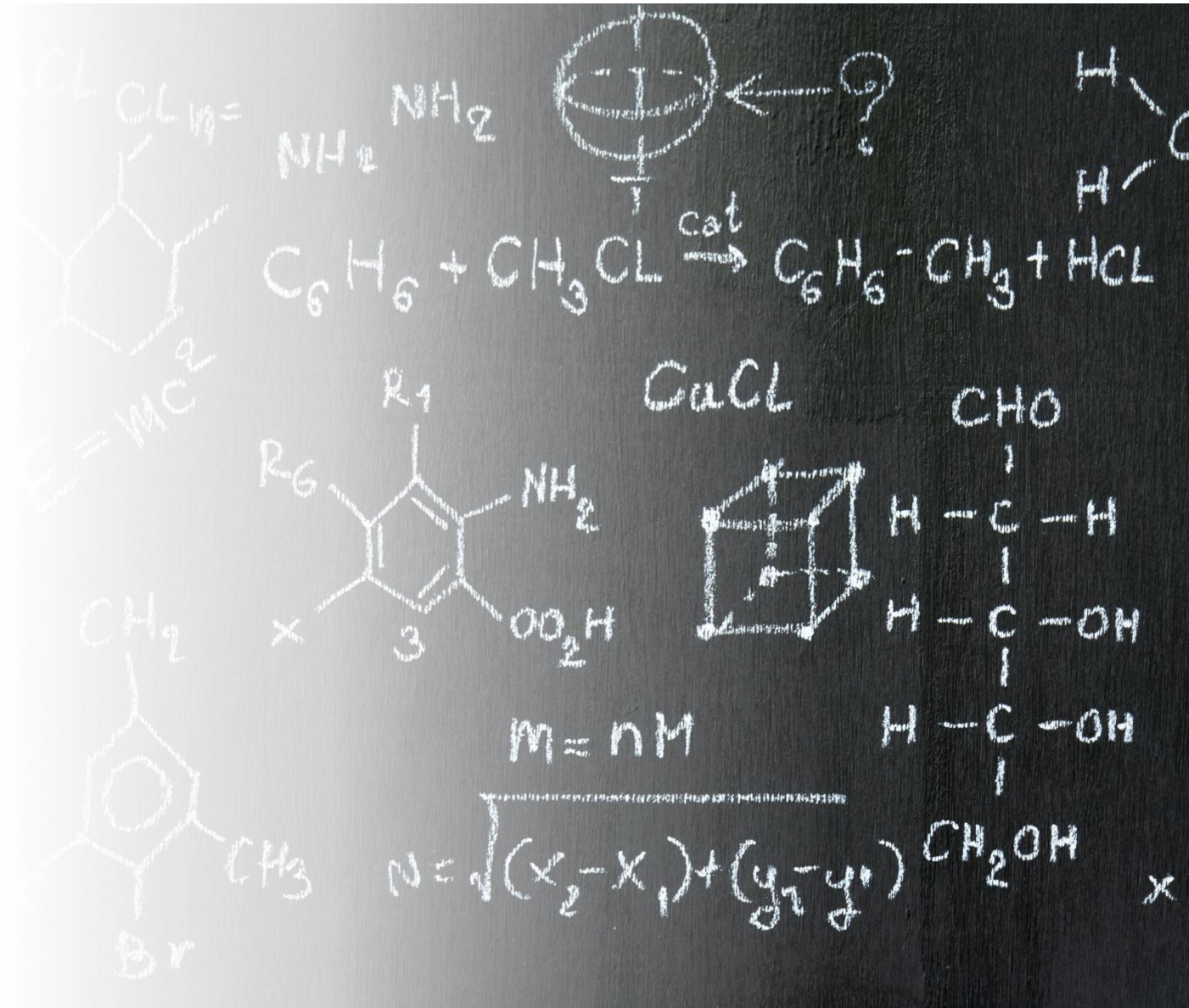
(2, 4) 0.4089220628888078

(3, 6) 0.8429263481500496

(3, 3) 0.5380289691033573

Page 10 of 10

- Duas formas principais:



BANANA.VECTOR

```
array([2.02280000e-01, -7.66180009e-02,  3.70319992e-01,  
       3.28450017e-02, -4.19569999e-01,  7.20689967e-02,  
      -3.74760002e-01,  5.74599989e-02, -1.24009997e-02,  
       5.29489994e-01, -5.23800015e-01, -1.97710007e-01,  
      -3.41470003e-01,  5.33169985e-01, -2.53309999e-02,  
       1.73800007e-01,  1.67720005e-01,  8.39839995e-01,  
       5.51070012e-02,  1.05470002e-01,  3.78719985e-01,  
       2.42750004e-01,  1.47449998e-02,  5.59509993e-01,  
       1.25210002e-01, -6.75960004e-01,  3.58420014e-01,  
      # ... and so on ...  
       3.66849989e-01,  2.52470002e-03, -6.40089989e-01,  
      -2.97650009e-01,  7.89430022e-01,  3.31680000e-01,  
      -1.19659996e+00, -4.71559986e-02,  5.31750023e-01], dtype=float32)
```

Outras Formas...



- *FastText*
- *Glove*
- *Bert*