

# NLP con FastAI

SaturdayAI - Kerri Rapes

# Programa

- Sobre mi
- Conversica
- NLP Basics
  - ◆ Que es NLP
  - ◆ Idiomas son difíciles
- Big Data vs Small Data
- Small Data
  - ◆ Stop Words
  - ◆ Stemming and Lemmatization
  - ◆ Regex

- NLP y Español
  - ◆ Poblaciones de hablantes nativos
  - ◆ Poblaciones futuros de hablantes nativos
  - ◆ Talento en las américas
- Big Data
  - ◆ FastAI - IMBD Classifier
  - ◆ IMBD Classifier - Español

# Sobre Mi



*“La gringa de gringolandia”*

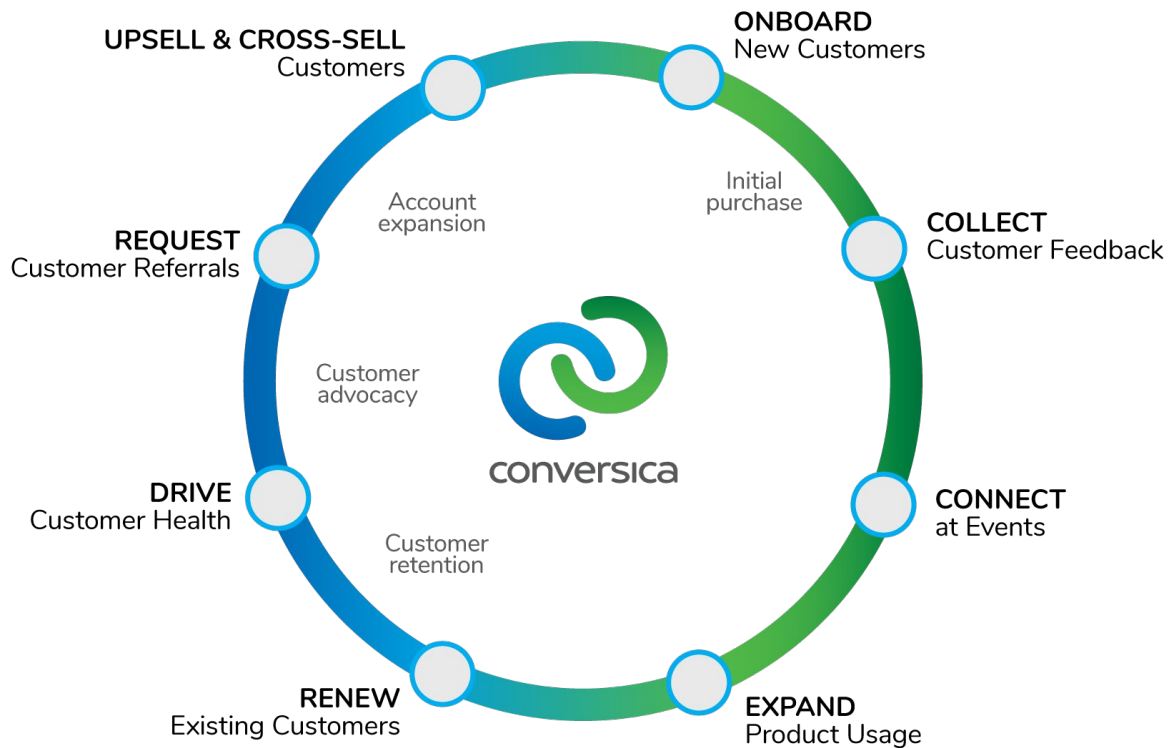
# Sobre Mi



Dynamic  
English

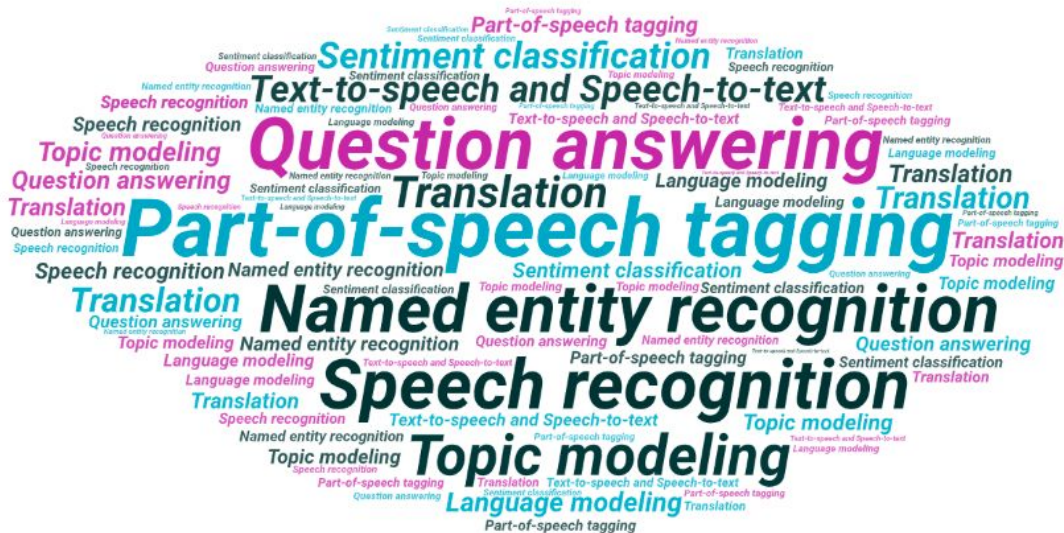
**Schlumberger**

# Conversica



# NLP Basics

Que es NLP?



1. Chat Bots
2. Interpretación automática de encuestas
3. Medición de la magnitud de los terremotos

# NLP Basics

Idiomas son difíciles?

The image displays two instances of the Google Translate interface, illustrating the process of machine translation for Natural Language Processing (NLP) tasks.

**Top Example:**

- Source Language:** English (selected)
- Target Language:** Turkish (selected)
- Input Text:** "She is a doctor. He is a nurse."
- Output Text:** "O bir doktor. O bir hemşire."
- Character Count:** 31/5000

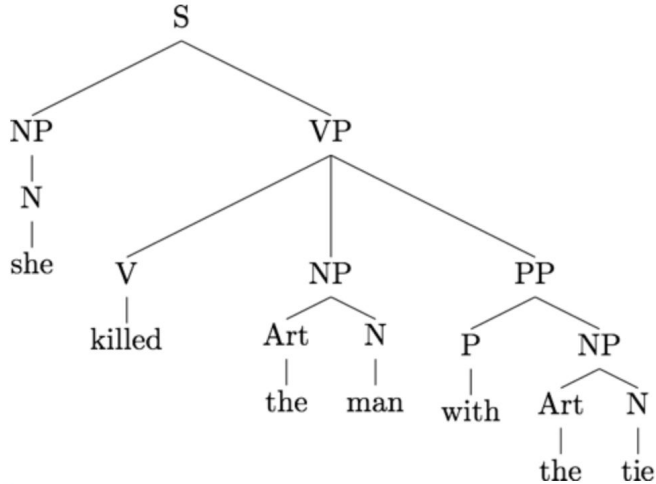
**Bottom Example:**

- Source Language:** Turkish (selected)
- Target Language:** English (selected)
- Input Text:** "O bir doktor. O bir hemşire"
- Output Text:** "He is a doctor. She is a nurse ✓"
- Character Count:** 28/5000

# NLP Basics

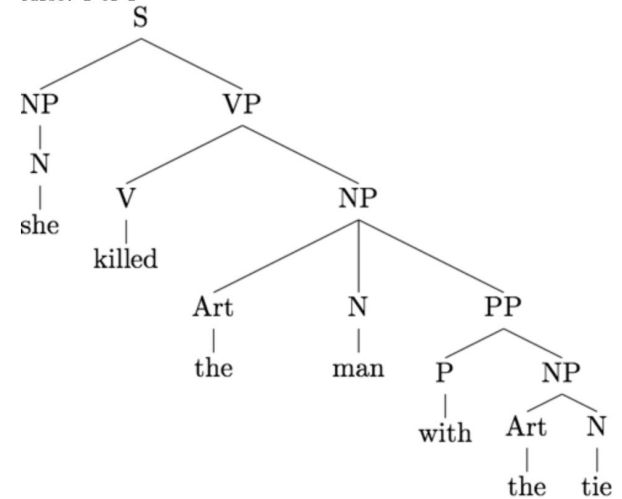
Idiomas son difíciles?

***She killed the man with the tie.***



Parse: 2 of 2

La corbata era instrumento usado  
para matar el hombre



El hombre tenía una corbata



# Big Data vs Small Data



SMALL DATA

# Small Data

## Stop Words



**Terrible Maps**  
@TerribleMaps

Follow

The most popular word in each state



8:06 AM - 18 May 2019

```
In [134]: vectorizer = TfidfVectorizer(input='filename', stop_words='english')
          dtm = vectorizer.fit_transform(filenames).toarray()
          vocab = np.array(vectorizer.get_feature_names())
          dtm.shape, len(vocab)
```


# Small Data


Regex - “Regular expressions is a pattern matching language.”

Problema: Identificar números del teléfono

- 123-456-7890
- 123 456 7890
- (123)456-7890
- 101 Howard

<https://regexone.com/>

 **RegexOne**  
Learn Regular Expressions with simple, interactive exercises.

 Interac

### Lesson 1: An Introduction, and the ABCs

**Regular expressions** are extremely useful in extracting information from text such as code, log files, spreadsheets, or even documents. And while there is a lot of theory behind formal languages, the following lessons and examples will explore the more practical uses of regular expressions so that you can use them as quickly as possible.

The first thing to recognize when using regular expressions is that **everything is essentially a character**, and we are writing patterns to match a specific sequence of characters (also known as a string). Most patterns use normal ASCII, which includes letters, digits, punctuation and other symbols on your keyboard like `%#$@!`, but unicode characters can also be used to match any type of international text.

Below are a couple lines of text, notice how the text changes to highlight the matching characters on each line as you type in the input field below. To continue to the next lesson, you will need to use the new syntax and concept introduced in each lesson to write a pattern that matches all the lines provided.

Go ahead and try writing a pattern that matches all three rows, **it may be as simple as the common letters on each line.**

Exercise 1: Matching Characters

Task	Text
Match	abcdefg
Match	abcde
Match	abc

Continue >

*Solve the above task to continue on to the next problem, or read the [Solution](#).*

# Small Data

## Stemming and Lemmatization

```
In [143]: from nltk import stem
```

```
In [144]: wnl = stem.WordNetLemmatizer()  
porter = stem.porter.PorterStemmer()
```

```
In [145]: word_list = ['feet', 'foot', 'foots', 'footing']
```

```
In [146]: [wnl.lemmatize(word) for word in word_list]
```

```
Out[146]: ['foot', 'foot', 'foot', 'footing']
```

```
In [147]: [porter.stem(word) for word in word_list]
```

```
Out[147]: ['feet', 'foot', 'foot', 'foot']
```

```
In [9]: import spacy
```

```
In [80]: from spacy.lemmatizer import Lemmatizer  
lemmatizer = Lemmatizer()
```

```
In [81]: [lemmatizer.lookup(word) for word in word_list]
```

```
Out[81]: ['feet', 'foot', 'foots', 'footing']
```



---

# NLP y Español



# NLP y Español

## Poblaciones de hablantes nativos

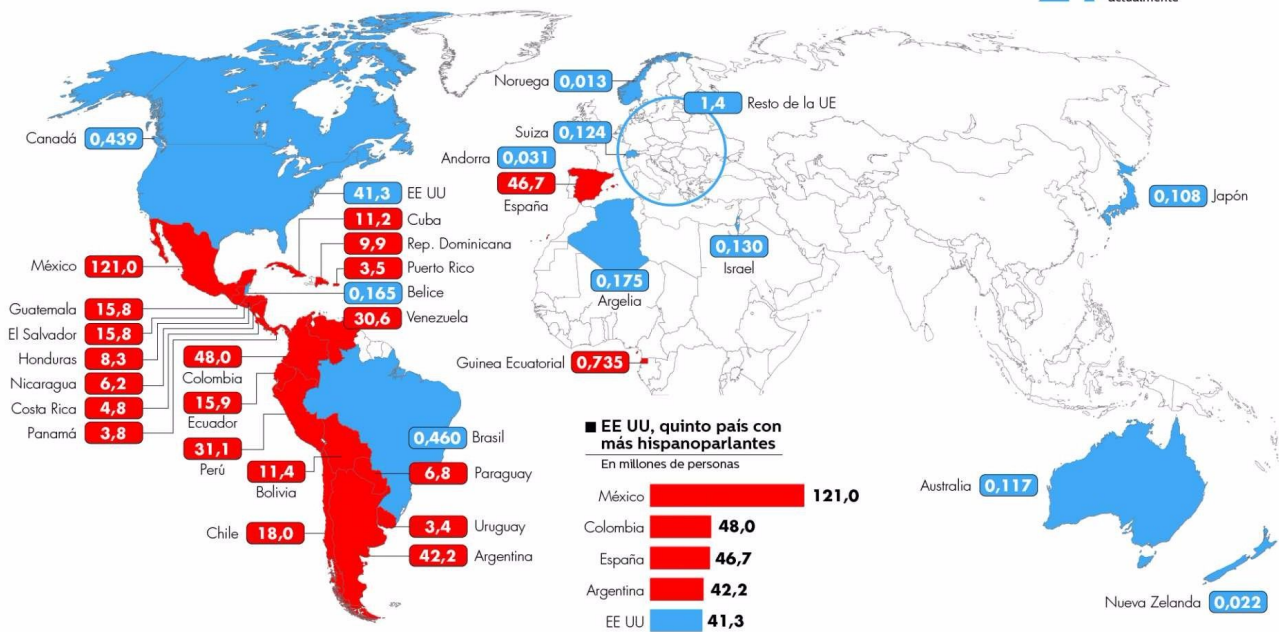
### El idioma español en el mundo

#### ■ Población hispanoparlante

En millones de personas ■ Español como lengua oficial ■ Lengua oficial distinta al español (principales países)

**470** millones de personas que hablan español de forma nativa

**68** millones de personas que hablan español sin ser su lengua nativa  
**21** millones de personas que estudian español actualmente



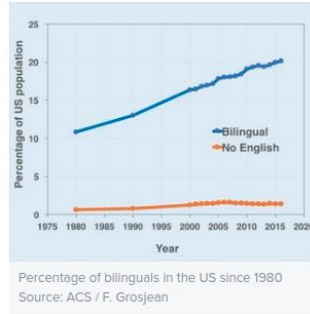
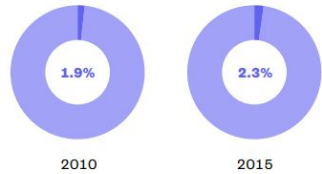


# NLP y Español

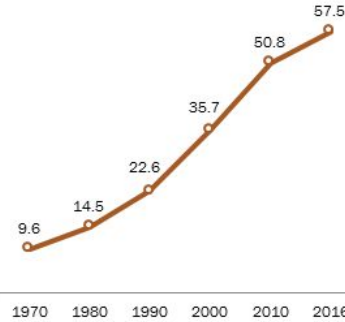
## Poblaciones futuros de hablantes nativos en EEUU

Employers were seeking **far more bilingual workers** in 2015 than they were in 2010.

FIGURE 3: SHARE OF ONLINE JOB LISTINGS FOR BILINGUALS



U.S. Hispanic population hits new high  
*In millions*

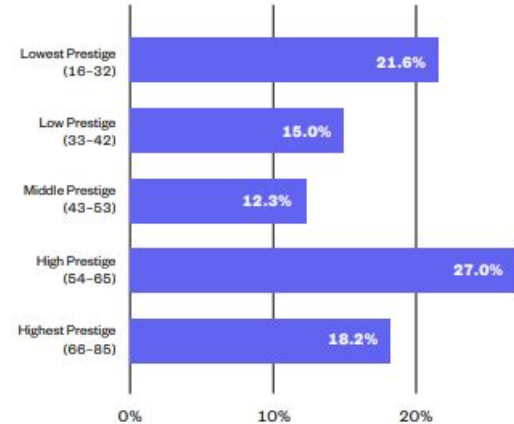


Note: 1990-2016 estimates are for July 1.  
Source: 1970-1980 estimates based on decennial censuses (see Passel & Cohn 2008). 1990-2016 estimates based on intercensal population estimates and Vintage 2014.  
PEW RESEARCH CENTER

[http://www.newamericaneconomy.org/wp-content/uploads/2017/03/NAE\\_Bilingual\\_V8.pdf](http://www.newamericaneconomy.org/wp-content/uploads/2017/03/NAE_Bilingual_V8.pdf)

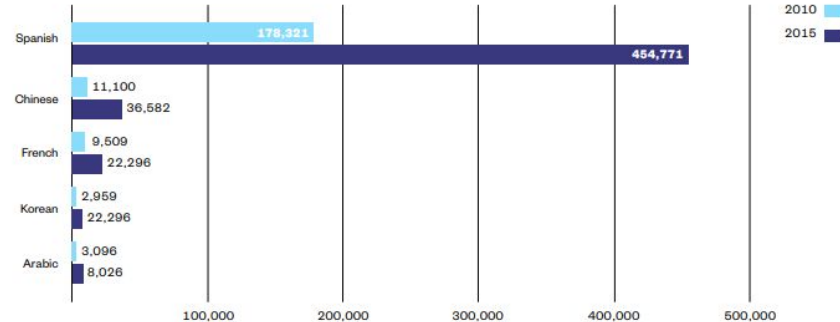
Demand for bilingual skills is not limited to one end of the skills-spectrum but instead is **spread across the economy** as a whole.

FIGURE 6: CHANGE IN SHARE OF JOBS TARGETING BILINGUAL WORKERS, 2010-2015



Source: Burning Glass Technologies, Labor Insight. Data pulled on April 13, 2016

FIGURE 10: NUMBER OF ONLINE JOB LISTINGS FOR WORKERS WITH BILINGUAL SKILLS IN SPECIFIC LANGUAGE



# NLP y Español

## Talento en las américas

- Chile (27th)
- EEUU (28th)
- Brasil (38th)
- Argentina (39th)
- Colombia (45th)

## Which Country Has the Best Developers?

Ranked by Average Score Across All HackerRank Challenges

Rank	Country	Score Index	Rank	Country	Score Index
1	China	100.0	26	Netherlands	78.9
2	Russia	99.9	27	Chile	78.4
3	Poland	98.0	28	United States	78.0
4	Switzerland	97.9	29	United Kingdom	77.7
5	Hungary	93.9	30	Turkey	77.5
6	Japan	92.1	31	India	76.0
7	Taiwan	91.2	32	Ireland	75.9
8	France	91.2	33	Mexico	75.7
9	Czech Republic	90.7	34	Denmark	75.6
10	Italy	90.2	35	Israel	74.8
11	Ukraine	88.7	36	Norway	74.6
12	Bulgaria	87.2	37	Portugal	74.2
13	Singapore	87.1	38	Brazil	73.4
14	Germany	84.3	39	Argentina	72.1
15	Finland	84.3	40	Indonesia	71.8
16	Belgium	84.1	41	New Zealand	71.6
17	Hong Kong	83.6	42	Egypt	69.3
18	Spain	83.4	43	South Africa	68.3
19	Australia	83.2	44	Bangladesh	67.8
20	Romania	81.9	45	Colombia	66.0
21	Canada	81.7	46	Philippines	63.8
22	South Korea	81.7	47	Malaysia	61.8
23	Vietnam	81.1	48	Nigeria	61.3
24	Greece	80.8	49	Sri Lanka	60.4
25	Sweden	79.9	50	Pakistan	57.4



HACKERRANK PROGRAMMER OLYMPICS (2016)



# Big Data

## FastAI - IMBD Classifier

### Problema:

*Vamos a ver críticas de películas de IMDB. Queremos determinar si una revisión es negativa o positiva, según el texto. Para hacer esto, utilizaremos el aprendizaje de transferencia.*



+



=



# Big Data

FastAI - IMBD Classifier - español

## Problema:

*Queremos determinar si una revisión es negativa o positiva, según el texto, pero no tenemos un base de datos equilibrado*

