**Current Developed NLP Tasks in SingularityNET**

Agst 10

**Services**

* Language Understanding – CNTK recurrent LSTM
* Named Entity Recognition – recoginize entities like PERSON, ORGANIZATION, and LOCATION from texts
* OpenNMT Romance Translator – for romance or European language
* Sentiment Analysis –
* Text Summarization – currently limited to domain of news articles
* Translation – between language parise. Currently limited to English to German only.
* Language Detection – Service using Polyglot
* Named Entity Disambiguation – using Wiki Dump
* Correference Resolution – using allennlps’s model
  + co-reference – means a task of finding or identifying all expressions or words that refer to the same entity in a text.

e.g Bill said he would come.

Proper noun Bill and pronoun he refer to the same person, namely toBill

**Course NLP**

Table of Contents

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 10](#__RefHeading___Toc629_51171938)

[WEEK 1 10](#__RefHeading___Toc626_51171938)

[Introduction to NLP 10](#__RefHeading___Toc631_51171938)

[State of the art of NLP in Language Technology 11](#__RefHeading___Toc633_51171938)

[Why is NLP so Difficult 11](#__RefHeading___Toc635_51171938)

[The Task of NLP is Very Difficult, so what tools do We need to Challenge this Difficulty ? 12](#__RefHeading___Toc637_51171938)

[Text Processing 12](#__RefHeading___Toc639_51171938)

[Regular Expression 12](#__RefHeading___Toc643_51171938)

[Regular Expressions in Practical NLP 14](#__RefHeading___Toc653_51171938)

[1. Word Tokenization 14](#__RefHeading___Toc657_51171938)

[2. Word Normalization 16](#__RefHeading___Toc659_51171938)

[3. Lemmatization 16](#__RefHeading___Toc661_51171938)

[Morphology 17](#__RefHeading___Toc663_51171938)

[4. Stemming 17](#__RefHeading___Toc665_51171938)

[Sentence Segmentation 18](#__RefHeading___Toc771_51171938)

Introduction video – <https://www.youtube.com/watch?v=GazFsfcijXQ&feature=youtu.be>

NLP is a field of Artificial Intelligence that deals with helping computers to understand, interpret and manipulate human language.

This course is designed for Absolute beginners with the Goals of helping them either

* Start NLP related Startup
* Do Consulting Work
* Find Full-Time NLP Related Work
* and for the one who wants to be masters of NLP

NLP is endlessely fascinating with topics like

* Sentiment Analysis
* Text Generation
* Dialogue System
* Translation
* and also there is so much that you can do with text data in 2019 and its truly incredible

Prerequiste for this NLP Curriclum are

* Basic Python
* Basic Linear Algebra
* Basic Calculus
* Basic Probability

If I am really Beginner I will take 2 months for this course 3-4 hours per day. But if you can take much more longer time.

Join Slack Channel to have discussion with your Journey and to keep you motivated to Complete this Course. But you know I have many motivation to complete and being masters of NLP.

Few days ago a team at Google Brain realesed XLNet a retrained model that outperforms the previous state of the art just a few months ago called **BERT.** That achieves state of the art results on 20 different tasks. Including Question Answering, Natural Language Inference, Sentiment Analysis, and Document Ranking.

Researchers really exicted that they finish BERT and Perfectly Integrate into their Pipeline when something new just like this broke state of the new art.

Such World of Machine Learning is Dank. Honestly I would not have to explain in any other way.

Dank is slang (Excellent, High Quality, also Unpleasantly damp and cold)

The Goal of this Course is to help you to learn

* Pre Deep Learning NLP techniques
* Deep Learning NLP techniques
* that you become very comfortable with Transfer Learning
* and give you a tips for Deep Reinforcement Leanring at the End.

But The most Important Part of this Curriclum is the **Transfer Learning** Bits.

If you Really dont have time to Learn anything alse and you just want to build something app ASAP. So just Directly Dive into Transfer Learning Portion.

Each of this Course Weeks has the series of

* Video Lectures
* Reading Assignments and
* Projects – to ensure you have understand the materials

Remember most Learning are occurred by Doing Not by Just Reading means just by passively absorbing information so the Projects are really important.

I have set **Pytorch** to be the library of the Choice for this Curriculum, since there are so many clean Pytorch NLP examples on Github thanks Facebook.



Btw Why Siraj Perefer Pytorch over Tensorflow, is its only the reason that many clean Pytorch NLP examples are available on Github. Is its only because of this reason because I doubt that may be Siraj has a got a money from facebook or have a relationship with facebook or Siraj may not be happy with Simple Google products because of his personal thoughts eventhoug the Tensorflow is really good than PyTorch. I don’t know because many people prefers Tensorflow than Pytorch even here in iCog.

Week 1

* Language or NLP Terminology and Preprocessing techniques.
* terminology(pragmatics, semantics, syntax, morphology)
* text preprocessing(stemming, lemmatization, tokenization, stopword removal)
* Project
  + using NLTK to perform Stemming, Lemmatization, tokenization, stopword removal on a dataset of your choice

Week 2

* Language Models and Lexicons (Pre-deep Learning methods)
* what is Lexicons
* Pre-deep Learning Methods (Statistical Language Model like HMM, Topic Modelling w LDA)
* Project
  + Building Hidden Markov Model for Weather Prediction in Pythorch

Week 3

* Word Embeddings (Word, Sentence and Document)
* Project
  + Implementing Word2Vec
  + Crreate dependency parser all in PyTorch

Week 4 and Week 5

* Deep Sequence Modeling
* Seq2Seq Models(Translation, Summarization, Question Answering)
* Attention based Models
* Deep Semantic Similarity
* Project
  + Developing a Translator and a Summarizer. All Seq2Seq models in Pytorch

Week 6

* Dialogue Systems
* Speech Recoginition
* Dialog Managers
* NLU(Natural Language Understanding)
* Project
  + Create a dialogue System using Pytorch and
  + a Task Oriented dialogue System Using DialogFlow to order Food

Week 7

* Transfer Learning
* how to build biomedical startup
* transfer learning with BERT/GPT-2/ELMO
* Project
  + Playing With Pytorch Transformers, Using it for one of 9 downstream task and compare their results.

Week 8

* Future NLP(Modern NLP Research Topics)
* Visual Semantics
* Deep Reinfrocement Learning
* Project
  + Policy Gradient Text Summarization.
  + There is an implemented models and then reimplement it in Pytorch

**WEEK 1 –**

Eventhough NLP makes up the majority of NLP state of the art algorithims. Ist important to start our Journey with Terminology before ML and I found a course that does this perfectly and it’s called “Speech and Language Processing” byProfessor dan jurafsky at Stanford.

Also video lectures in the playlist so watch up until 2.5 they cover important NLP reminology terms like Pragmatics, Syntax and morphology.

And there’s a whole theory of language that linguists have devoted their lives to studying before statistical analysis came along and we should pay homage to it by gaining some domain knowledge of language theroy before beginning to model it for specific tasks.

And Lets’ not forget **Pre-Processing** the video lectures cover all the different preprocessing techniques to perform when working with text data which is usually very messy, lemmatization, stop word removal, tokenization,

rarely do we find a corpus that is perfectly ready to be fed into a model we have to clean it up first.

The reading Assignments for this topic is –

* from the previous Professor’s book of the same title are sufficient, read the first two chapters

**As for Project**

Project for this Week is to Perform three different Pre-Processing techniques of your choice on a paragraph of text you pull from the web or create by yourself with the NLTK.

Before We start using PyTorch We can Use NLTK to clean up text data and this first week will help you master that skill.

**Week 2: Language Models and Lexicons**

For this Task, Don’t Worry Deep Learning is already Coming, but there were decades of research in NLP that have nothing to do with deep neural networks that reserachers have been using that we should learn about. It’s important to understand our history in order to figure out where we want to go next. One of the methods that previous researchers have been using for this Language Models and [Lexicons ] is Statistical Language Model.

Statistical Language Model is a probability distribution over a sequence of words.

If we can create a langauge model from some text corpus we can use it to perform tasks like text classification or text generation, in the style of that corpus it captures the essence of it.

**Materials**

The video lectures for this week comes from the University of Washington’s courts on NLP lectures 2 through 6 cover language models, text classifiers, hidden markov models and parsing techniques.

The associated readings for each lecture are really good read 4 through 10. skip chapter 5 since that one talks about neural nets and we are holding off on that until later.

Lets’ also fit a technique called LDA(Latent Darish Lay Allocation) into here by Reading the provided tutorial blog post. LDA is a technique that lets you generate the topic from a corpus and it’s called topic modeling.

**As for the Project**

* use Pytorch to build a popular statistical technique called a Hidden Markov Model, I have an example for you to look at for help on this one but use your HMM to classify texts from a data set accordingly.

What about Lexicons here ?, because he do not speak any thing about Lexicons

**Week 3: Neural Networks and Word Embeddings**

In all Modern NLP techniques use Neural Networks in Some way or form so it’s very important to understand how this Statistical Architecture works.

We can use neural networks to create embeddings. Embeddings are mathematical representations of words, sentences, and some times even whole documents. and these embeddings allow us to compare texts, visualize it and improve the accuracy of newer models and prevent warfare(waraana). Kidding about last one.

**Materials**

Standford has the perfect course on how to get started with this embeddings, it’s called NLP with Deep Learning. video lectures 1 through 5, from Word Embeddings up to Optimization Startegies fit this topic nicely and it already got a list of suggested reading for each video lecture as well as notes.

**As for the Projects**

* The associated assigments of above Standford course are pretty nice, go through and complete all three but just the coding parts and do them all in PyTorch. These assignments will have you visualize and implement word to vec models and
* then create a dependency parser all in pytorch.
  + Dependecy Parsing is a task of analyzing the syntactic or grammatical dependency structure of words in a given input sentence S. or Establishing a relationship between “head” words and words which modify those heads in a sentence.

The output of a dependency parser is a dependency tree where the words of the input sentence are connected by typed dependency relations.

Input neurons --- hidden neurons --- output neurons

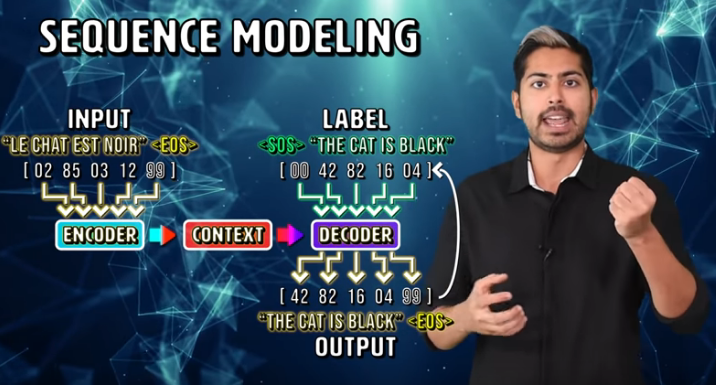
This is a Neural Network. It makes a mistakes and It learns from them. Be like a Neural Network.

**Week 4 – 5 Sequence Models**

Now we’re going to dedicate two weeks to the next module(Sequence Modelling) just because there is so much involved with Sequence Modeling wether it be an Audio Waveform of your voice or a sentence. We can think of this type of data as sequenctial since it consists of related data points that occur sequentially like words or waves.

When we feed a seqeunce let’s say of words into a model and ouput a new sequence say a new sentence of words we can consider that sequence to sequence modelling.

So many tasks in NLP can be formulated as such Seq2Seq model e.g machine translation, text summarization, question answering and the Encoder/Decoder model that takes in and outputs a sequence of words has been the go to model for this type of modeling



There are Many variants of Seq2Seq Models out there and

* many sorts of techniques like how to apply attention to certain layers with different papers like attention is all you need
* many sorts of variants of Recurrent Neural Networks like LSTM and GRU networks

**Materials**

Week 4 of Deep Learning Coursera Course on NLP covers all sorts of sequence modelling techniques particularly Machine Translation.

For the Reading Lectures checkout chapter 10 of the Deep Learning Book by Ian Goodfellow on Sequence modelling

Also I wanted to Fit the DSSM(Deep Semantic Similarity Model) into here as well.

DSSM are able to model how similar to text strings are and have a wide variety of use cases in popular tasks. The provided blog will bring you up to speed pretty fast on this.

**As for Projects**

Now the bulk of the work from this week will come from the Project

* Using the popular Movie lens data set that contains movie dialogues, then Build two different models a text summarizer and a question answering system in PyTorch.

This is a crucial project the reason it’s important to implement recurrent sequence models using pytorch for this task is so that you will have a deep appreciation for the state of the art today in sequence modelling.

Which actually has nothing to do with Recurrent Network variants today it’s all about the Transformer Model. We will dive to depth about Transformer Modeling in week 7

Yes Deep Learning requires GPUs it’s okay if you dont’ have one just do it in a Web Browser with Google Colab with their free compute or Folydhub.

**Week 6: Dialogue Systems**

this weej we will see how to develop the Dialogue Systems like Alexa(Microsoft), Google home and Siri(Apple) .

This is the study of how humans interact with the machines and all the different ways we can architect these types of dialogue based systems

****

**Materials**

the last week of the same coursera course called Dialogue systems has the perfect set of video lectures for this covering different dialogue system designs that help classify user intent and send the appropriate response or follow-up question to the user.

For Reading Assignments chapter 24 of the Speech and Language Processing book will do great

Currently this is an active area of research in different Companys.

Google recently demoed its **duplex** product that was able to automate a phone call with a human in a surprisingly realistic way.

Google Duplex – is a new conversational systems from Google that is currently live in the majority of the US and it allows certain users to make a reservations at restaurants. By making similar realistic calls just like human throug phone.

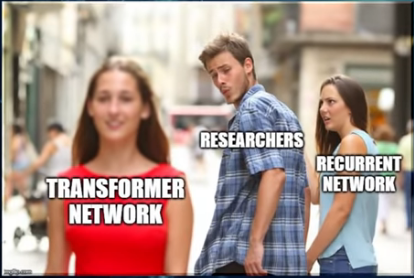
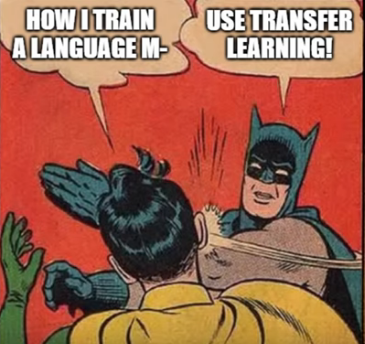
Think abut How Google AI wants to make your life easier.

**As for Project**

There is one project here Create a Dialogue system using a PyTorch that enables some one to check the Weather. That means it has to take context into account each question depends on the previous response text.

**Week 7: Transfer Learning**

this is the most important Week in fact if you really wanted to get started with NLP like make something impressive in just a few hours just start here.



Recent models like **BERT** are all free for any one to download and use in their app as a starting point for training on a specific task like Sentiment Analysis or Text generation.

I have a great video on how BERT works for both theoretically and in practice in the form of an app . I also have a video on GPT too.

These models are generally all variants of what’s called the transformer network.

**Transformer Network** is the architecture that’s now used by all the major research labs instead of Recurrent Networks for Sequence Learning. that’s huge and it’s not yet part of any major curriculum because it’s so new.

**Materials**

* I’ve got some great reading assignments for you on this front.
* Sebastian Reuter has a great blog post on why NLP just had its imagenet moment and what that means and two other blogs that describe in technical detail how the transformer network works in the context of all of these different pre-trained models.
* I’ve also got giant list of pytorch pre-trained examples

**As for Projects**

your project for this week is to pick two of these models and use them both for the sentiment analysis for product reviews which will involve transfer learning.

get used to the idea that you have to retrain an existing giant model on some data, specific to your task. This process is much faster than doing it from scratch with Recurrent Neural Network.

**Week 8: Modern NLP Research**

first lets get Vision Language hybrid Models out of the way. By that I mean anything having to do with translating between an **image** and **text.** like image captioning, visual question answering or even text to image. This involves combining sequence models with convolutional nets and module 6 of the Edx microsoft NLP course will give you some great video lectures on this.

Now to Deep Reinforcement Learning

this is pretty experimental, first check out this hilarious(extremely amusing) yet educational blog post by Goldberg called an Adversarial review of adversarial Generation of Natural Langauge.

He is basically saying that you can’t just throw some hot new technique like adversarial networks at a problem that requires a deep understanding of langauge theory and expect it to beat the state of the art.

With that being said there is interesting work being done applying deep RL2 language. This carnegie mellon lecture will give you a great primer and so will module 5 of that the same Edx course

Reinforcement Learning applied to static text data is really interesting to me because it’s usually applied to scenarios where time is a crucial element like games, real-world supply chains or sensor data.

There is a policy gradient model that’s used for text summarization I found implemented in tensorflow. the last project is to merely re-implement that in Pytorch this will enable you to more fully grasp this concept of combining the function approximation capability of deep neural networks with reinforcement learnings mathematical framing of a time-based continuous environment to more efficiently solve an objective function

and that’s it. And after finishing this Curriculum I want to see you do one of three things

1. Start a Startup
2. do consulting work
3. find a full time nlp related work.

Knowledge and action are inextricably(in way that is impossible to separate) connected so act, use knwoldge to act in this world for the betterment of your own life in the lives of others in this way AI technology will help you find happiness, Wealth, meaningfulness and inner peace.

Remember to believe in your ability to learn. You can learn NLP , you will learn NLP, and if you stick to it, eventually you will master it.

I hope you find this video useful and if you did please click the red subscribe button.

Good Luck Wizard’s(xonqaayi, man with magical power) Im rooting for you.(ani siifan dhaaba jira)

# **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **WEEK 1**

## Introduction to NLP

Hi I’m dan jurafsky and Chris Manning and We are very happy to welcome you to our course on Natural Language processing.

Current time is particularly exciting time to be working on natural language processing. The vast amount of data on the Web and social media have made it possible to build fantastic new nlp applications

lets look at some of them

1. Questin Answering: IBM’s Watson

* won Jeopardy challenge on February 16, 2011
* jeopardy is an american televesion game show, its format is a quiz competition in which contestants(person who takes part in competition) are presented with general knowledge clues in the form of answers, and must phrase their responses in question form

1. Information Extraction
   * from the sent email automatically recognize that the email is schedule or Event and then notice the dates, time, and place of the schedule and add this schedule to your calendar.
2. Sentiment Analysis

* from a bunch of reviews of a given product e.g camera on the internet and then automatically determine from the reviews that what people care about in this product(camera) attributes e.g they want to know if its has good zoom or affordability or size and weight.

e.g if the reviewer says nice and compact to carry that’s a positve sentiment. But if the reviewer says the camera feels flimsy, then automatically we determine that its negative sentiment.

Then we can aggregate the negative and positve sentence for each attributes of the camera because a given camera can have positive sentiment for zoom and quality but negative sentiment for size and wieght or affordabilitiy

1. Machine Translation

* can be fully automatica
* can help human translators
  + in online translators from one language to another it can predict the next word coming

### State of the art of NLP in Language Technology

1. Well Developed and Mostly Used

* Spam detection. e.g lets go to usa (OK), buy v1agra..(Spam)
* Part of Speech tagging. from sentence identify N, V, ADV, ADJ,
* Named Entity Recogintion. From sentence identify Person, ORG, Location

1. Making Good Progress

* Sentiment Analysis
* Coreference Resolution
* Word Sense Disambiguation
* Parsing
* Machine Translation
* Information Extraction

1. Still really Hard to Develop

* Question Answering(QA). Asking question of medicine
* Paraphrase e.g XYZ acquired ABC company yesterday. Means similar to ABC has been overtaken by XYZ, but its difficult for machine to deduce like this.
* Summarization. Taking bunch of news article and generate short useful information
* Dialog. Making a dialog what a movie is about

### Why is NLP so Difficult

1. Ambiguity Problem

* this ambiguity problems are called “Crash blossoms” → ambiguosly worded headline whose meaning can be interpreted in the wrong way. Or that have multiple interpretation.

e.g Red Tape Hold Up New Bridges

can have two meaning 1, Red Tape Delay building of New Bridges or 2 Red Tape Supports building of New Bridges. This is called Word Sense Ambiguity

* ambiguity is not only found on some part of the words or only in headline(title). Ambiguity is pervasive means its present or noticeable in every part of the sentence.
* e.g Fed raises interset rates. Can be

1. where main verb is **raises** Fed, raise the rate of an intereset

2. where main verb is interest, it means Fed raises is interested in rates.

1. Non – Standard English
   * especially text written in social media like facebook, twitter, just random campitalization, unusual use of words(like you by U), # tags, user ids and so on
2. Segementation Problems

e.g the New York-New Heaven Railroad.

For this sentence how can we know the correct segementation(separation) is

[the] [New York]-[New Heaven] [Railroad]. Since there is also “-” in the middle of York-New how can we segement just like above as human. Because this is not sth lik “in-law”

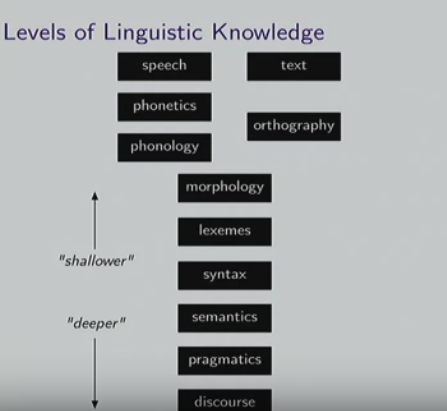
1. Understanding idioms
   * e.g dark horse, get cold feet
2. neologisms
   * new words that have never seen before. e.g retweet,
3. tricky entity names
   * even if that movie name is an english words it may be two or three words and its really difficult to know where the name of that movie starts in a sentence

### The Task of NLP is Very Difficult, so what tools do We need to Challenge this Difficulty ?

* Knowledge about langauge
* knowledge about the World
* a Way to combine this knowledge source

**What does it mean to “know” a language ?**

Surprise question his native language is English, and he is NLP teacher and suddenly he ask his students by saying “do you think I knew an English”​? Its really a difficult questions.



Speech and Text are very shallow from knowledge of Languages.

But Linguistic have developed very different kinds of Study about Languages

**e.g** for Text side they developed study called **Orthography**, which is a writing systems. Different languages have different kinds of writing systems.

In English we have only 26 characters, you need to use whitespaces

In chinese its totally different because it have tens of thousands of characters and there is NO whitespace at all.

On Speech side we have

Phonetics – that study the way we make a sound or speech or phonemes

and set of phonemes that avialable in different language is different.

This Class of NLP by Standford is mainly focus on to teach Key theory and methods for **Statistical NLP.** Algorithim like

* Viterbi
* Naive Bayes, Maxent clasifiers
* N-gram Language Modelling
* Statistical Parsing
* Inverted index, tf-idf, vector models of meaning

## Text Processing

The Most basic tools and Fundamentals for Text Processing is the Regular Expression

### Regular Expression

* Regular expression is a formal language for specifying text strings.

e.g how can we search for any of these ?

Woodchuk, woodchuk, woodchuks, Woodchuks

* so to deal with such kind of problem we have tools like

Regular Expression: Disjunctions → brackets []

* Disjunctions is the simplest and fundamental tools in Regular expression.
* Letters inside square brackets [] can match any of letter in that brackets. e.g [wW]oodchuck –> can match Woodchuk or woodchuk. [0123456789] – can match any digit
* Ranges [A-Z] → can matches all capital letters from A to Z. [0-9], [a-z], [A-Za-z]
* e.g you can use <https://www.regexpal.com/>
* Negations, using carat(^) → it means we do not need or want this one.

e.g [^Ss] Neither S nor s, [^A-Z] – Not an upper case letter at all,

Regular Expression: More Disjuntion → pipe ( | ) either … or

* e.g groundhog | woodchuck. Either groundhog or woodchuk
* a | b | c → either a or b or c, and its the same as [abc]
* we can combine brackets and pipe e.g [gG]roundhug | [Ww]oodchuck

Regular Expressions: ? \* + .

* ? → letters or characters before question mark is optional e.g colou?r → color | colour
* \* → called kleene star, it matches 0 or more characters before it e.g oo\*h → oh | ooh | ooooh
* + → called kleene plus, it maches 1 or more characters before it e.g o+h → oh | ooh | ooooh
* . → shows any characters can replace it e.g beg.n → begin, begun, beg3n, begzn

Regular Expressions: Anchors ^ $

* ^ → mathes the beginning of the line or sentence, e.g ^[A-Z] all capital letters at the beginning.
* $ → matches all the end of the line or sentence e.g [A-Z]$ – all capital letters at the ends.

\.$ → matches all periods at the ends of the line or sentence. Since . Shows any character in regexp so to get only period we use \.

.$ → matches all characters and letters at the end of the line or sentence

The Process of this Regular Expression that we went through was based on two kinds of Errors

1. False Postives(Type 1 Error) → matching strings that we should not have matched
2. False Negative(Type 2 Error) →Not maching things that we should have matched

So it turns out that in Text Processing or NLP we are always dealing with these two kinds of Errors.

1. Increasing accuracy or precision (minimizing false positves)

2. Increasing coverage or recall (minimizing false negatives)

Regular Expressions play a surprisingly large role in Text Processing.

But for many hard tasks, we can use machine learning classifiers, but even for this regular expressions are used as features in the classifiers, because they are very useful in capturing generalizations.

### Regular Expressions in Practical NLP

Lets see a regular expression in Standford English Tokenizer

Text Normalization

Text Normalization is an important task of Text Processing.

Every NLP task start by Word Tokenization, then Word Normalization, then Stemming and Lemmatization.

But before word Tokenization we need to segment a text into list of sentences.

### 1. Word Tokenization

* Segmenting/tokenizing words in running text
* token means just single word in a sentence, but its little bit different from actual word by some definitions and characters. **e.g**

How many **words(Tokens)** and **types**(**Vocabulary**) are there in the following sentences. Words is just the total number of words. Type the total number of unique words from the vocabulary. In words double word can be counted as two words, but in types(vocabulary) double words counted as one word also word that contain the same word forms or lemmas counted as one word. In types(vocabulary) our focus is in lema or baseform of the word from the vocabulary.

* “I do uh main – mainly business data processing”. This is complicated because of words like “uh” and “main” or “mainly”. We call things like “uh” as filled pauses and things like “main -mainly” fragments. Especially this is a big issue in speech recognition.

What about cat and cats in a sentence is they are two words or one words, how machine counts them as one. So basically for Tokenization we have Two approaches Lemmatization and Stemming.

Also for “they lay back on the San Fransisco grass and looked at the stars” how many words and Type are there in this sentence. If we count San Fransisco as two words we end up with 13 tokens but if we count it as one word we end up with 12 token. Also for the type we have 10 or 11 types if because the word “the” appears double, they, their are the same word forms or lemma and we can count them as one word type.

|  |  |  |
| --- | --- | --- |
|  | Tokens = N | Types = V |
| Switchboard Phone Conversations | 2.4 million | 20 thousand |
| Shakespeare | 884, 000 | 31 thousand |
| Google N-grams | 1 trillion | 13 million |

See in Google N-grams there are 31 million different unique(distinct) vocabulary, so how many vocabulary words are there in English. You see the number of vocabulary words in language is very very huge and its keeps going increasing and increasing each day because new names are being created each day for each new developed things in the world.

There are some standard tools for Text Processing like

* unix tools tr
* NLTK

**Issues in Tokenization**

how we can tokenize the sentence it words is its only splitting them based on space only, No Tokenization has much more duty than Splitting based on sentence. So we have to decide the standard what gonna we use when we change words into Tokens.

**e.g** how can we tokenize the following sentences into words or tokens ?

Finland’s capital → Finland | Finlands | Finland’s ?

What’re, I’m, isn’t → What are, I am, is not

Hewlett-Packard → Hewlett Packard, state of the art ?

Lowercase → lower-case | lowercase | lower case ?

San Francisco → one token or two ?

m.p.h, PHD → ???(in English issue with a peroid becomes a huge issues so we have to know how we can tokenize such like words)

**Note:** Tokenization is not Splitting with Space. Its far far more complicated beyond that.

**The issue in Tokenization can even more complex in other languages**

* in French e.g L’ensemble → one or two token ? L | L’ | Le
* in German, Long nouns are not segmented as in English. e.g ‘Life Insurance Company Employee” means in German “Lebensversicheerungsgesgesllschaftsangestellter”, which is spelled as one single word, so for such kinds of language we need to have compound splitting.
* In Chinese and Japanese, we have a different problem, there is no space between the word at all. So we need to have smart tokenization algorithim for such kind of languages. And another issues in japanese also there are different kinds of alphabets in the same sentence like japanese alphabet, katakana alphabet, hiragana alphabet, kanji alphabet, romaji alphabet. So for such language we need to have smart tokenization algorithims

1. **Lemmatization**
2. **Stemming**

### 2. Word Normalization

* Normalizing means just changing all the texts or words in a sentence in to the same form or formats. So in preprocessing Normalization is also the most important features that we need to do. **e.g** In Information Retrieval; indexed text & query terms must have same form. We want to mach **U.S.A** and **USA**
* so it means just implicitly define equivalence classes of terms. e.g deleting periods in a term.
* Case folding means changing all the letters into lower case, but here also we need to make some exceptional. e.g upper case in mid-sentence ? Fed vs fed, SAIL vs sail. General Motors. For such kind we need to make exceptional.
* For Machine Translation, Sentiment Analysis, Information Extraction, and others Case is very important. e.g when you change all text into lower case for the advantage of preprocessing there is a big difference between ‘US’ and ‘us’. The Government of US is going to release new rule. When you change to small letters machine can recognize us as us not as US(United State).

### 3. Lemmatization

* We also want to do Lemmatization to reduce words into the same form as in vocabulary or base form. E,g am, are, is → be car, cars, car’s, cars’ → car
* Generally task of lemmatization is just finding the correct dictionary headword form.
* We can see Lemmatization as also task of Normalization because its also just change into base word form.
* And its very important for most of applications particularly for machine translation

So this Generally leads us to Topic of Word called **Morphology**

### Morphology

* Its the study of morphemes and
* **Morphemes** is the study of smallest meaningful units that makes up the Words.

There are two kinds of morpheme

1. **Stems** → the core meaning bearing units in a word
2. **Affixes** → is the bits and pieces that adhere to stems, often with grammatical functins.

Affix is just a set of letters added to the beginning or end of a root word(stem) to modify its meaning. Generally there are two types of Affix, Prefix(added to the beginning) and Suffix(added to the end)

e.g in word cars – “car” is stem and “s” is affixes

### 4. Stemming

* Stemming is just a task of preprocessing that takes words and removes affixes from it and returns the root or stem of that word.
* Stemming is just crude chopping of affixes
* Stemming is **language dependent**. Because in each language the way affixes added to the root word is different from each other. So each language needs to have its own Stemming Algorithim. You cant use English Stemming Algorithims for other language.
* So for each task of NLP whether its Statistical Methods or Deep Learning Methods, We Need Text Preprocessing methods like Lemmatization, Stemming to reduce Vocabulary size and increase the efficiency of the models. So to do this preprocessing task we need to Study Morphology of each language because its different from one language to another. Particulary in Machine Translation we have to be series for this task.

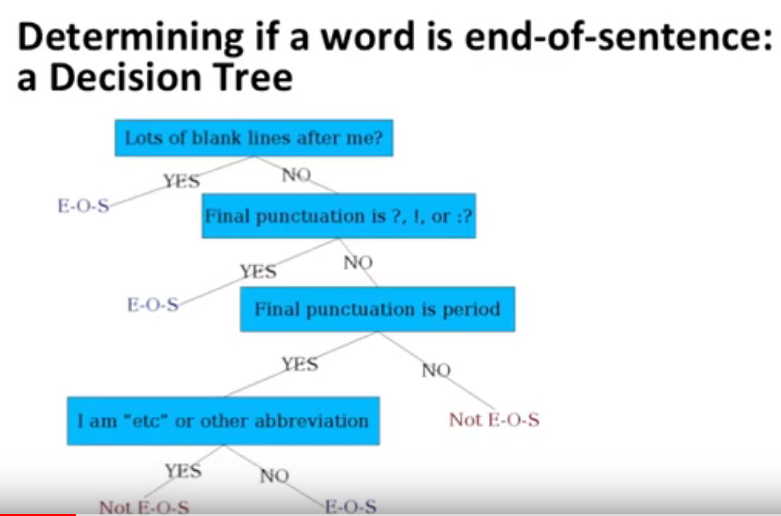
**e.g** in english automate, automates, automatic, automation → automat

* The most common English Stemmer is → Porter’s Algorithim, this algorithim is just a series of rules. like Step 1a, Step 1b, Step 2(for long stems), Step 3(for even longer stems)
* in english language the stemming algorithim have to be smart when removing “-ing” suffixes from the words just because it was comes at the end.
* e.g for coming, having, going, loving, its ok but for nothing, something, cunning its not suffixes but its part of that word. So the algorithim have to take consider for this one.

This is simple example of morphology but in some language there are much more complex morphology and requires complex morpheme segmentation. e.g in Turkish the ff sentence   
“Uygarlastiramadiklarimizdanmissinizcasina” is a single words and its meaning in english is “behaving as if you are among those whom we could not civilize”. So just like in German this one Turkish is also complex problem, so Stemming algorithim for this Turkish also needs to be much more smarter.

### Sentence Segmentation

* The most important task in Text Processing is also Sentence Segmentation, how we can segements sentences from running texts . How can we know the end of sentences in a text and split or segments them.
* in English sentences that ends with punctuation marks like !, ? are relatively good because they are unambiguous and they show end of a sentences
* But period(.) is quite ambigous because in English sentence period(.) has multipurpose task, like
  + showing Sentence boundary or end of sentence
  + Abbreviations link Inc. or Dr.
  + Numbers like .02% or 4.3
* so we cant assume that Period(.) is an End of sentence, so what we can do to solve the period problem is build a binary classifier that looks at a period and decides is am at end of sentence or not. And to build this classifier we can use hand written rules, regular expressions or building machine learning classifier
* the simplest classifier for this is the Desicion tree.



**More sophisticated decision tree features**

checking case of word before or after with period(.) whether its Upper, Lower, Cap or Number, because this can give us much information about that period whether its end of sentence or not.