

Applied Natural Language Processing
Prof. Ramaseshan Ramachandran
Department of Computer Science and Engineering
Chennai Mathematical Institute, Madras

Lecture - 07
Preprocessing

(Refer Slide Time: 00:15)

PREPROCESSING

- ▶ Corpus creation ✓
- ▶ Modifications required to make the text suitable for processing ✓
- ▶ Understanding the problem is very crucial in choosing a preprocessing steps
- ▶ Preprocessing steps are unique to a problem ✓
- ▶ Improper preprocessing schemes may lead to loss of lexical context → 30%
Regular Exp.

NLTK

30%

Regular Exp.

NPTEL

In our introduction session, we saw a very high-level overview of what we are going to be doing in the next 12 weeks, to start with we are going to be looking as the Preprocessing as the first step. any corpus that is given to you, you know you need to really look at what kind of corpus there is. We need to understand the domain in which the corpus is available, without understanding the domain it is not easy for you to do any of the tasks; especially in the preprocessing, you need to really know what you are going to be doing ok. Every task requires different sets of preprocessing steps.

So, first is about creating the corpus, creating the corpus is very simple. Just start collecting all the relevant documents in one folder and then either created in the binary form or some form and then make it available for you to process it. make it machine-readable, nowadays you do not have to really worry about the corpus if you really want to do certain experiments, those corpora are available. If really if you look at various APIs they provide a certain corpus for you to play with ok. And, then we need to do certain processing preprocessing to make sure that the text is suitable for our application.

So, what kind of processing that we do in the text we will talk about that little later. again as I mentioned understanding the problem is very very critical in terms of choosing the preprocessing step. we will show a few examples and then see why I am talking about this and you can say that as preprocessing steps are very unique to a given problem. every problem is different and every problem requires different sets of preprocessing steps. If you do not do the processing well, I think you are going to be losing a lot of lexical content that will result in very poor application value.

So, this is one of the most critical aspects of natural language processing, in many cases, it is about 30 percent of the application generation. if you have about 5 months of job or 1 year of job you know 30 percent of your work would be on the preprocessing side. It is a very very crucial step ok. I am not going to be talking about a certain application that you may have to use or the APA that may have to use to do the preprocessing, I am going to be assuming that you know those ok.

For example, a regular expression you may have to read it to yourself, if you do not know this. And, understand how regular expressions can be used to preprocess texts or extract some information from the document or replay certain values using search and replace more and so on.

(Refer Slide Time: 03:51)

COMMONLY USED PREPROCESSING STEPS FOR ENGLISH

Preprocessing consists of (a) tokenization, (b) normalization and (c) substitution

- ▶ Case folding - Convert all text into lower case
- ▶ Stemming - running → run
- ▶ Lemmatization - best → good
- ▶ Remove misspellings
- ▶ Punctuations
- ▶ white space, newline, tabs...
- ▶ Removing contractions - isn't → is not, I'd → i would
- ▶ Remove scripts, form variables, for HTML and XML
- ▶ Tokenization

Handwritten notes: buy bright buy, Speed 5 km/hr → m/sec

NPTEL

So, the basic steps involved are one is tokenization; tokenization, as I mentioned earlier, is about splitting the text into words right and then the second step is normalization and

then the third one is the substitution. you probably may not have substitution in many cases, but certain preprocessing certain applications require a substitution as well. we will talk about each one of those and what they are. this you know normalization is about you know I mentioned earlier that if you have bought.

So, these two will be converted into buy or in some cases suppose in if you are solving a problem physics problem, where the speed is mentioned as, and then the normalization step requires you to convert this into meter per second. you need to be able to read this, understand this and then convert the unit into this and appropriately change this number into the required number for meters per second. this is useful in certain cases, this may not be useful in every application that you are dealing with ok.

And this is where you know this substitution the key a km per hour will be substituted by a meter per second and then the numerical value also would be replaced by the appropriate value here. that is where substitution comes ok.

(Refer Slide Time: 06:09)

COMMONLY USED PREPROCESSING STEPS FOR ENGLISH

Preprocessing consists of (a) tokenization, (b) normalization and (c) substitution

- ▶ Case folding - Convert all text into lower case
- ▶ Stemming - running → run
- ▶ Lemmatization - best → good
- ▶ Remove misspellings
- ▶ Punctuations
- ▶ white space, newline, tabs...
- ▶ Removing contractions - isn't → is not, I'd → I would
- ▶ Remove scripts, form variables, for HTML and XML
- ▶ Tokenization

Handwritten notes on slide:

- buy, bought, buying
- Monday, REC, h → <S>
- Buy buy
- 1) t = 5 seconds
- 2) t = 5 s
- 3) t = 5 sec

Or in those cases for example, if the time is represented as 5 seconds and then this is in document 1, in document 2 t is represented as 5 s, in document 3 it is represented as t equal to 5 sec. for you to normalize this what you do is you pick up some standard representation of seconds and then convert each one other into that standard representation. For example, the standard representation for second to second to s as

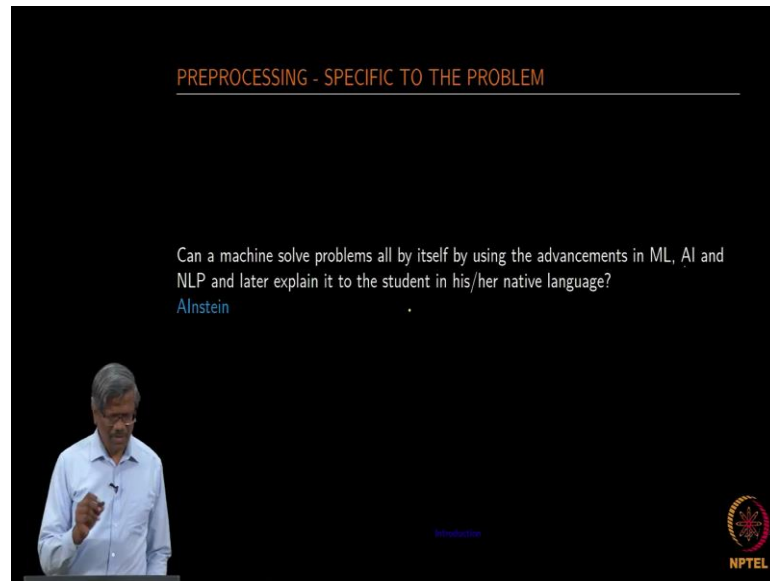
some notation that I am using. all would be replaced in the standard form. you normalize it and then substitute this using the standard notation.

So, these are the three operations that you would normally find in the preprocessing steps. we would be using something called case folding where we convert all text into lowercase. we do not want to be distinguishing this and this as two different words ok, if you do the comparison. we want to convert all the text into lowercase and then we want to do this stemming part. Stemming is nothing, but you know taking of these suffixes.

And then lemmatization is a process that converts best into good ok. And, then misspellings if there are any wrong spellings available as part of the corpus you replace that. Take out all the punctuations and then remove unnecessary white spaces, tabs, newline etcetera ok. Remove contractions for example, is not it is replaced by is not and I would be replaced by this ok. Removal of scripts, if you are dealing with HTML corpus you should be removing all these JavaScript's, form variables and some XML tokenization so on ok.

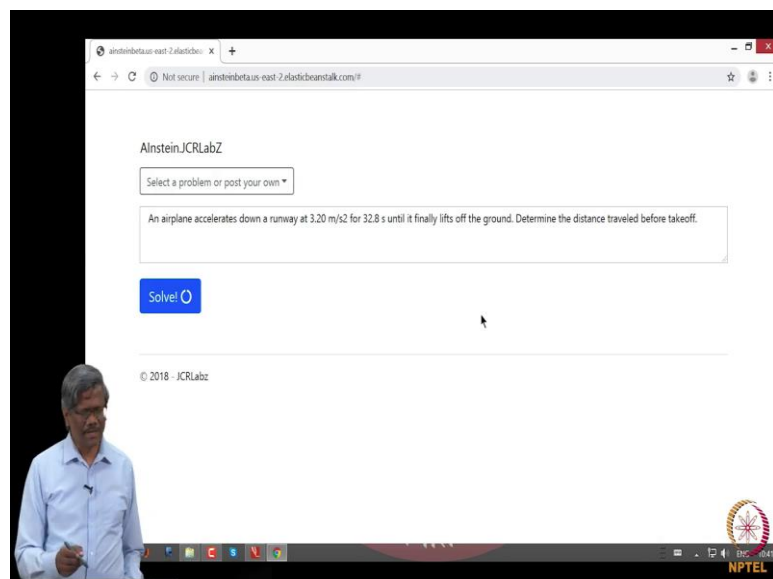
So, I do not need any of those tags to be present as part of the content because for me the content is important not just the tags. In some cases the tags provide you have high-level information for example, if you have the bold and then some content it could be the title. you want to retain that in certain cases. you want to be very sure about what you want from the content and then write your preprocessing step. as I mentioned earlier it is not an APA that we can just use you know ok, preprocess my content; it will do all the job on its own. No, you have to pick up your own preprocessing steps and then perform this. this is a very crucial step in any natural language processing.

(Refer Slide Time: 09:27)

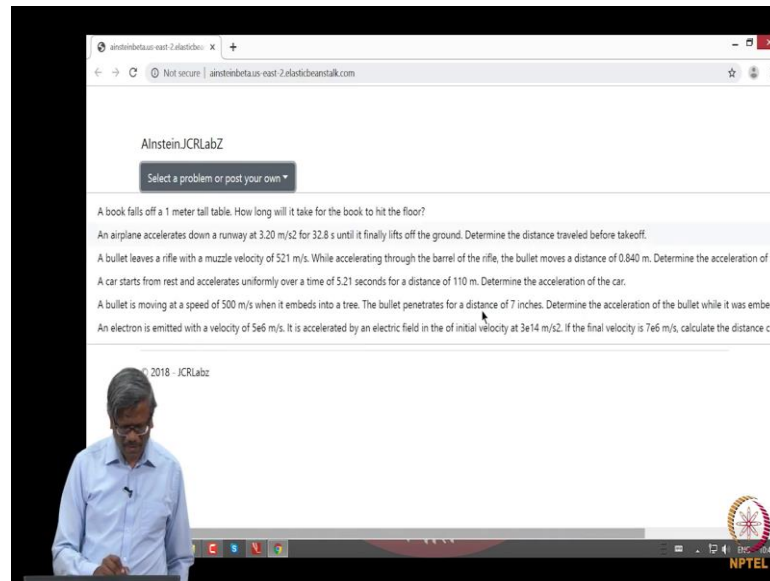


To give you an example of how we would do the preprocessing, I am just going to quote some small examples; the example would be from physics ok. Can the missions all problems all by itself by using the advances in machine learning, artificial intelligence and NLP and later explain it to the student in his or her native language ok. I will just give one example of that, I hope it works ok.

(Refer Slide Time: 10:04)

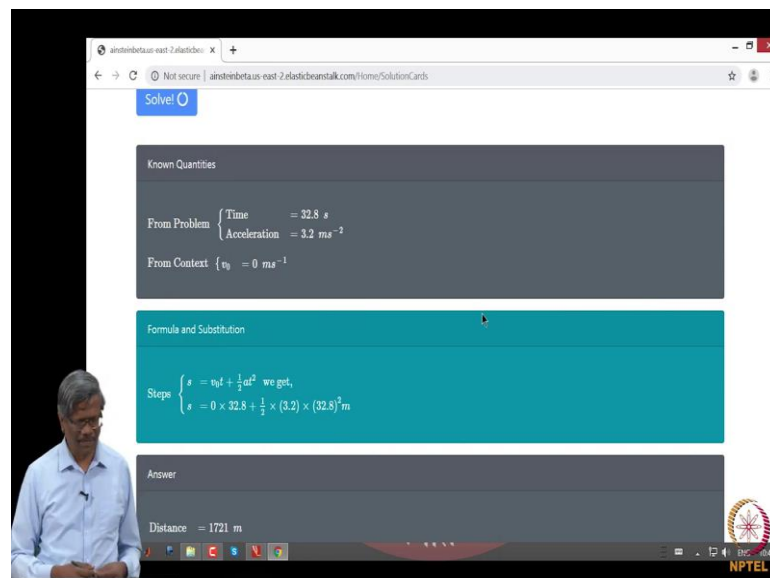


(Refer Slide Time: 10:09)



So, this is one application that reads let us say a problem of this type. A plane accelerates down a runway at 3.2 meters per second squared and so on right. This is a problem, I just want to post the problem and the machine should solve this problem on its own by looking at the problem. Finding out what are all the given values in terms of you know here we have an acceleration, we have our time and then it lifts off the ground and then distance determines the distance traveled before.

(Refer Slide Time: 10:47)



So, we want the machine to read this natural language text, take out all the known quantities on its own like this. for example, from the problem it says time is given. it is normalized to look at this right as I mentioned earlier the problem gives 3.20 ms^{-2} *Type equation here*. Say it understand that it could be meters per second square, it could be the acceleration. it translates that into the normalized form, the same fashion 32.8 is the time, and then it from the context it understands that it is starting right from the rest. the initial velocity is going to be 0 meters per second ok.

(Refer Slide Time: 11:41)

Known Quantities

From Problem $\begin{cases} \text{Time} & = 32.8 \text{ s} \\ \text{Acceleration} & = 3.2 \text{ ms}^{-2} \end{cases}$

From Context $\begin{cases} v_0 & = 0 \text{ ms}^{-1} \end{cases}$

Formula and Substitution

Steps $\begin{cases} s = v_0 t + \frac{1}{2} a t^2 \text{ we get,} \\ s = 0 \times 32.8 + \frac{1}{2} \times (3.2) \times (32.8)^2 \text{ m} \end{cases}$

Answer

Distance = 1721 m

2018 - JCRLabz

NPTEL

And, then it grows and then looks at all these values and then tries to find out which formula it fits into. And, then replaces those values in the formula, replaces those symbols with the values and finally, it computes the distance right. it computed the distance as of 1721. assume that we have an application like this, actually, this is an application that solves, you may want to try this if you want to go to this website. You can type some problems from the kinematics and then see whether it solves the problem ok.

So, you can solve certain problems from the kinematics, it is a prototype you know. I warn you that it may not be able to solve every problem in kinematics. To show you an example of what we can do with natural language processing I am just showing this. for me to do this task right, I need to do some preprocessing. It is not enough if I just use the tokenization alone, it is not enough if I just do this stemming alone. It is not enough if I just convert this into the text into small letters, I need a lot more than that right.

(Refer Slide Time: 13:05)

AINSTEIN

A kangaroo is capable of jumping to a height of 2.62 m. Determine the takeoff speed of the kangaroo.

Classifier
The most important step - to find out which problem class the text belongs to.
Meta-knowledge:
Class = Kinematics + Agent: Gravity

Question Resolver
Independently identifies possible questions from the meta-knowledge. It matches with the problem question and determines the question type - Direct or Indirect.
Meta-knowledge:
Class = Kinematics + Agent: Gravity
Acceleration = -9.8 m/s²
Final velocity = 0 m/s
Height = 2.62 m
Direction = up
Search Speed = 1 m/s

Context Clues
Core reading engine identifies the context and suggests missing information that needs to be used to find the solution.
Meta-knowledge:
Class = Kinematics + Agent: Gravity
Acceleration = -9.8 m/s²
Final velocity = 0 m/s

Formula Identifier
It finds the right formula from the known quantities. The formula must fit the need. All quantities except those already known are known.
Meta-knowledge:
Class = Kinematics + Agent: Gravity
Acceleration = -9.8 m/s²
Final velocity = 0 m/s
Height = 2.62 m
Direction = up
Direct equation
Search Speed = 1 m/s

Known Quantities
Units, numbers and SI units are combined. Water refers to distance and height, 'u' is distance.
Meta-knowledge:
Class = Kinematics + Agent: Gravity
Acceleration = -9.8 m/s²
Final velocity = 0 m/s
Height = 2.62 m
Direction = up

Solution
Takeoff speed = 7.33 m/s

NPTEL

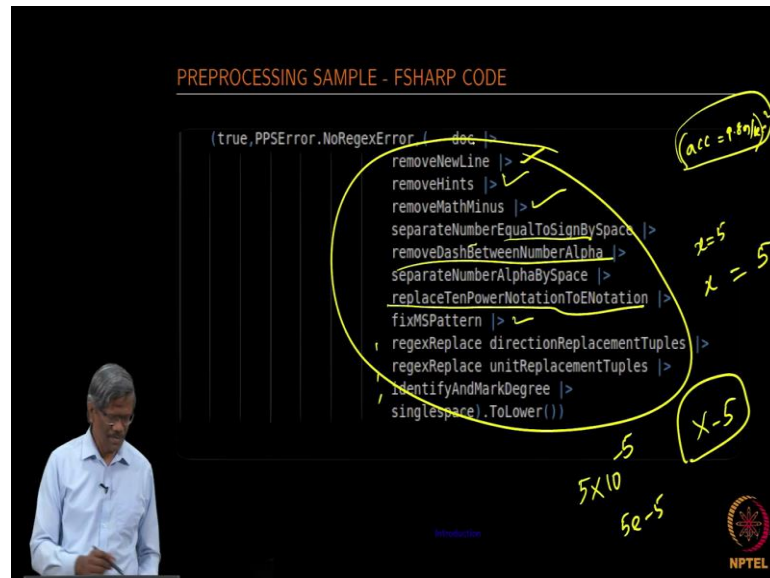
So, let me show you what these are certain tasks that the application does internally. It tries to find out whether it belongs to the kinematics or not first by looking at certain keywords that are found in the document. And, then it careers the meta knowledge and starts constructing that on its own. It tries to figure out some context, then I am just giving a very high level or I am not going describe the de details of what it does. It tries to understand from the problem what are the context clues; the context clues are about the acceleration against gravity and so on. This is related to the problem that you find here, it's not the one that I had shown there.

And, then it tries to find out what are the known quantities given in this problem statement, it says the known quantity is only this nothing else is given ok. it tries to figure out since it is taking off it is going to be against gravity. it gets some acceleration due to gravity and then the final velocity is going to be 0 because, it jumps and then stops and then drops back to the ground right. the final velocity is 0, it has seen and then if the height is given, the direction is in the upward direction and then tries to find out what the question is determine the take-off speed.

So, using those phrases it tries to find out what is the actual question and then it says its a direct question and it needs to find this. Since, it is the speed it also says this is a question mark a meter per second and then it tries to find out based on what it found and stored as meta-knowledge. And, then tries to map this with a formula and finally, figures out there

is a formula available and all the variables that are available could be fitted into that and if speed could be found and finally, it finds the speed.

(Refer Slide Time: 15:17)



So, for us to do this we require various preprocessing steps ok. I need to remove the unnecessary new line that is available as part of that. And then remove hints for example, certain problems will provide you a hint, the acceleration due to gravity equal to 9.8 meters per second square right. it does not want to look at that and get confused so it removes that ok.

Just assume that all those things that are happening are on the earth and then it removes certain symbols that are available as part of the original text. A separate number equal to a sign by a space for example, if x equal to 5, just write this as x equal to 5 in some way. And, then if there is something like this it removes those. replace TenPowerNotationToENotation. For example, if somebody gives 5 into 10 power minus 5 it replaces those into this form ok. that is the normalization that we do and replacement that we do and then fixes certain patterns in terms of the units and so on.

So, if you look at this you know there are so many preprocessing steps that are performed on this particular problem statement that we have in hand right. before we really take the problem to the machine to solve it, these are the set of tasks that you have to do as part of the preprocessing. And, as I mentioned unless we understand what the corpus is all about, in terms of its structure, in terms of the domain. It is not possible to

do any of this you know, we could only do certain tasks and it may not be really useful for the mission to use that ok.

(Refer Slide Time: 17:32)

CORPUS

- ▶ An airplane accelerates down a runway at 3.20 m/s^2 for 32.8 s until it finally lifts off the ground. Determine the distance traveled before takeoff
- ▶ How far will a car travel in 25 min at 12 km/h ?
- ▶ A jalopy with an initial speed of 23.7 km/h accelerates at a uniform rate of 0.92 m/s^2 for 3.6 s . Find the final speed and the displacement of the jalopy during this time
- ▶ A college student wants to toss a textbook to his roommate who is leaning out of a window directly above him. He throws the book upwards with an initial velocity of 8.0 m/s . The roommate catches it while it is traveling at 3.0 m/s [up]. a) How long was the book in the air? b) How far vertically did the book travel? A car accelerates in a straight line from rest at the rate of 2.3 m/s^2 . What is its final velocity after 55 m ? What is its time?
- ▶ 4. What height will a dart achieve 7 seconds after being blown straight up at 50 m/s ?

mpb m/s m s⁻¹

NPTEL


So, to give you some examples this is the original corpus the content that we have ok, I have given about 4 samples ok. you see the variations right here somebody will give you smh; that means, kilometer per hour you know you need to standardize this notation, 25 minutes should be converted into second.

So, you need to understand this is a time right and then the units and the number always occur in pairs in the problem statements that you will notice right in many cases. we need to be able to read them in pairs and understand that and then the notations would vary; some will say MPs, some will write ms, some will write ms minus 1. we need to be able to understand this when this occurs and then based on some preceding words earlier should really understand that, it is unit is representing the velocity and then convert this into the normalized form.

(Refer Slide Time: 18:48)

PROCESSED CORPUS

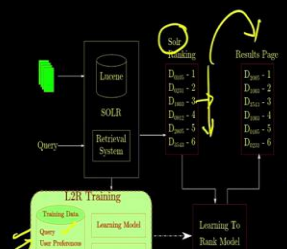
- ▶ An airplane accelerates down a runway at 3.20 m/s^2 for 32.8 s until it finally lifts off the ground. Determine the distance traveled before takeoff
- ▶ How far will a car travel in 25 minute at 12 km/h ?
- ▶ A jalopy with an initial speed of 23.7 km/h accelerates at a uniform rate of 0.92 m/s^2 for 3.6 s . Find the final speed and the displacement of the jalopy during this time
- ▶ A college student wants to toss a textbook to his roommate who is leaning out of a window directly above him. He throws the book upwards with an initial velocity of 8.0 m/s . The roommate catches it while it is traveling at 3.0 m/s [up]. a) How long was the book in the air? b) How far vertically did the book travel? A car accelerates in a straight line from rest at the rate of 2.3 m/s^2 . What is its final velocity after 55 m ? What is its time?
- ▶ 4. What height will a dart achieve 7 s after being blown straight up at 50 m/s ?




And, this is what the normalized representation after the preprocessing step. I have one more which I am not showing here, you can see the preprocessed steps in this case here got it.

(Refer Slide Time: 19:15)

HTML PREPROCESSING



1. Convert HTML to text →
2. Case folding - convert content to lower case →
3. Remove scripts →
4. Tokenize →
5. Term Frequency → -
6. Extract Vocabulary → -
7. Remove stop words - not for Language modeling →
8. Stemming/Lemmatization →



If you want to do the same thing on an HTML corpus for example, you have a help document that it's available as part of the product and its about 20 different products that you have. And, those documents are available as part of HTML text right and you want to do you want to provide quick help to the user by making it available in the cloud. And,

then you need to provide a search engine that searches through the document and brings the right content in front of the user by looking at all the HTML pages that are available as part of the corpus.

So, for us to do certain NLP operations we need to convert the HTML into the text because, this could contain a lot of images, Java scripts, form variables all that right. we need to convert that first into a text format without really losing the content of the source. And, then do the case folding, remove scripts, tokenize or do the term frequency, extract the vocabulary, remove stop words, do stemming or lemmatization or both and so on. this is for a task where we want to rank the document they are coming in from the document repository.

So, there we want to create a learning model that will try to rank the documents accordingly based on user preferences. For example, if I use Solr, Solr is one of the open-source applications that will help you in terms of indexing documents. It also gives you the list of documents available with respect to its own ranking ok. based on the frequencies of the words that are occurring, Solr gives you a certain ranking in this order. But, that ranking need not be always the right ranking you know if with respect to the user, the user may think that the document that is third in this list could be the most important one. he would have clicked or she would have clicked this several times during the search process.

So, the machine should be able to read, understand the user intent as well and then include that as part of the ranking and then create a modified ranking. that is what we call as learn to rank process, we will talk about this is one of our classes later. in the learning to rank I require all these frequencies, vocabulary information, bi Grams, trigrams information and so on, in order for me to understand this along with the query that is coming into the document. The user preferences, how many times they have clicked and then the ranking from this Solr, the results of the Solr; I combine all of those and then create a new learning model and then finally list that.

So, in order for me to do that my preprocessing steps are very different from what I had shown earlier right. the reason why I am showing this is every preprocessing step and everyone is very unique to a given problem so that we should keep this in mind. if you take another problem, so study the problem very well, look at what you are really going

to achieve in that application and then accordingly pick up your preprocessing steps ok.
with this I conclude the preprocessing and then we will continue with the rest of the
sessions later.