UofT 3666 – Applied NLP Final Project

Group Members: Linda Peto and Rahim Jiwa

Purpose

- The goal is to conduct a feasibility study to explore and demonstrate tools that be used to support your climate change application.
- We focused on three different areas:
  - PDF extraction
  - LDA topic modelling
  - Identifying Actions.

Data

- Data used were the Climate Change Docs and the Action CSV initially provided.
- In addition, we self generated and obtained data to enrich provided data.
- Added:
  - manually labelled 388 sentences from the Climate Change Docs as Non-Actions
  - Approximately 20 additional Climate Change PDFs.
- Additional files can be found on GitHub.

PDF Extraction

- Three different tools were explored: PdfMiner, PyPDF2, Tika.
- We found that each tool provided a different outcome. We provided a notebook comparing each of the tools against each other on the same document.
- Overall, each of the tools that we tested were easy to use and implement.
- Tika was the best performing. PDF Miner was the best pure Python option.
- There were a number of challenges that were faced:
  - PDFs formats can vary significantly from file to file. This increases the challenge of preprocessing the data for the various different cases.
  - There was a tendency to have poor transcriptions.
  - When batch converting PDFs there are cases that you need to be careful of: encryption, blank pages.

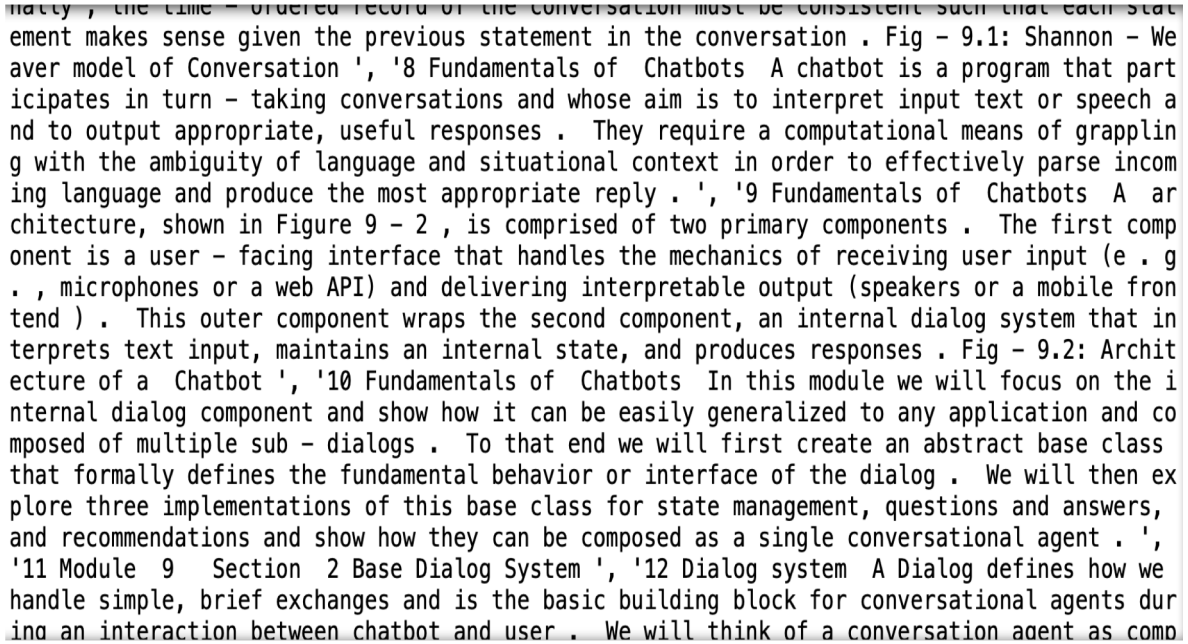
PDF Extraction - PDF Miner

- Printed output of the text is fairly clean.
- One issue with PDF Miner is that it can be sensitive to document names.
- Another issue that can be seen is that some sentences are cut and split into two or three lines. This has the potential to be problematic depending on the context.

```
Fundamentals of Chatbots

• A chatbot’s architecture, shown in Figure 9-2,

is

comprised of two primary components.
– The first component

is a user-facing interface that handles the
mechanics of receiving user input (e.g., microphones or a web API)
and delivering interpretable output (speakers or a mobile frontend).

– This outer component wraps the second component, an internal dialog
input, maintains an internal state, and

interprets text

system that
produces responses.

Fig-9.2: Architecture of a Chatbot

9
```

PDF Extraction - PyPDF2

- The output is a list of pages.
- When parsing the document, you end up having to go page by page.
- The raw output initially was messy. There were new line characters in between every token.
- Interesting aspect is dealing with encrypted files. For our purposes, we checked for encryption and skipped the encrypted documents.



```
nally , the time - ordered record of the conversation must be consistent such that each stat
ement makes sense given the previous statement in the conversation . Fig - 9.1: Shannon - We
aver model of Conversation ', '8 Fundamentals of  Chatbots  A chatbot is a program that part
icipates in turn - taking conversations and whose aim is to interpret input text or speech a
nd to output appropriate, useful responses .  They require a computational means of grapplin
g with the ambiguity of language and situational context in order to effectively parse incom
ing language and produce the most appropriate reply . ', '9 Fundamentals of  Chatbots  A  ar
chitecture, shown in Figure 9 - 2 , is comprised of two primary components .  The first comp
onent is a user - facing interface that handles the mechanics of receiving user input (e . g
. , microphones or a web API) and delivering interpretable output (speakers or a mobile fron
tend ) .  This outer component wraps the second component, an internal dialog system that in
terprets text input, maintains an internal state, and produces responses . Fig - 9.2: Archit
ecture of a  Chatbot ', '10 Fundamentals of  Chatbots  In this module we will focus on the i
nternal dialog component and show how it can be easily generalized to any application and co
mposed of multiple sub - dialogs .  To that end we will first create an abstract base class
that formally defines the fundamental behavior or interface of the dialog .  We will then ex
plore three implementations of this base class for state management, questions and answers,
and recommendations and show how they can be composed as a single conversational agent . ',
'11 Module  9   Section  2 Base Dialog System ', '12 Dialog system  A Dialog defines how we
handle simple, brief exchanges and is the basic building block for conversational agents dur
ing an interaction between chatbot and user .  We will think of a conversation agent as comp
```

PDF Extraction - Tika

- The output is a string.
- Easy to use.
- Output here has new lines and tab characters stripped out.
- Requires an up to date Java Server.
- Parses the whole document at once. This makes it easy to use, however, it can be problematic if you only need specific pages or have a pdf document with a blank page.

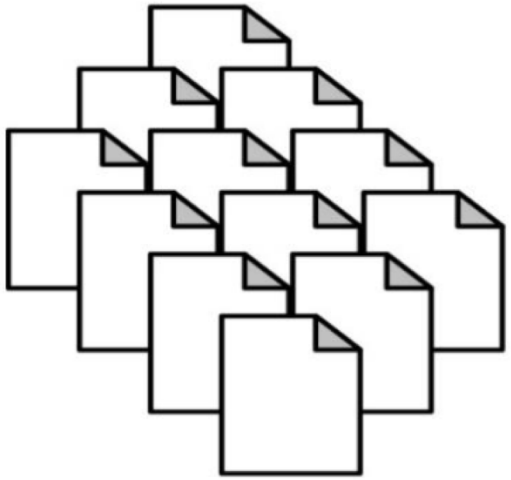
a conversation, a participant can either be listening or speaking. Effective conversation r equires at any given time a single speaker communicating and other participants listening. – Finally, the time-ordered record of the conversation must be consistent such that each st atement makes sense given the previous statement in the conversation. Fig-9.1: Shannon-Wea ver model of Conversation 8 Fundamentals of Chatbots • A chatbot is a program that part icipates in turn-taking conversations and whose aim is to interpret input text or speech a nd to output appropriate, useful responses. • They require a computational means of grappli ng with the ambiguity of language and situational context in order to effectively parse in coming language and produce the most appropriate reply. 9 Fundamentals of Chatbots • A chatbot's architecture, shown in Figure 9-2, is comprised of two primary components. – The first component is a user-facing interface that handles the mechanics of receiving user inp ut (e.g., microphones or a web API) and delivering interpretable output (speakers or a mobi le frontend). – This outer component wraps the second component, an internal dialog system that interprets text input, maintains an internal state, and produces responses. Fig-9.2: Architecture of a Chatbot 10 Fundamentals of Chatbots • In this module we will focus on the internal dialog component and show how it can be easily generalized to any application and composed of multiple sub-dialogs. • To that end we will first create an abstract base c lass that formally defines the fundamental behavior or interface of the dialog. • We will then explore three implementations of this base class for state management, questions and a nswers, and recommendations and show how they can be composed as a single conversational a

LDA Topic Modelling

- LDA (Latent Dirichlet Allocation) is an unsupervised method of generating topics present among a set of documents.
- The model takes a set of documents (which are made up of a collection of words), and examines the words within it, in order to determine the topics.
- It also suggests that each document is made up of a distribution of small topics. The idea here, is that it is possible to have multiple topics within the same document. LDA looks to see how likely a topic can be seen within a document.

LDA Topic Modelling

The document of a corpus comprise a number of topics



A topic is a distribution over words.

aioli fry pancake
kobe cream garlic

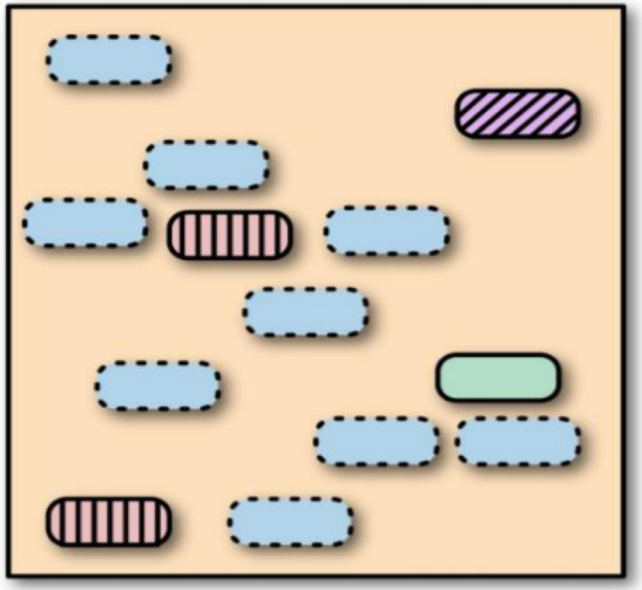
election stump run
president Iraqi

kobe nba record
draft finals tendon

note record pop
cello harmony trio

fair console game
unity tactical

A single document invokes multiple topics.



LDA Topic Modelling

- To perform LDA on this dataset, we tried using two different tools: sklearn and gensim.
- Initially, we started with sklearn as it was the package our group was more familiar with.
- We found that sklearn could implement LDA, however, it was not as robust as gensim.
- An advantage of sklearn is that it is a popular package, with superb documentation and lots of available guidance.
- Gensim is primarily focused on topic modelling, it provides greater functionality and is a better choice for a more focused NLP task.

LDA Topic Modelling

- With the Gensim package
  - Implementing the LDA model
  - Calculating the coherence scores.
  - Testing a random document on with the model
- Our best performing model had a Coherence Score of -0.13 utilizing the u_mass measure.
- Upon inspection our topics also seemed to fit. The terms are related, however, within climate change, it is hard to discern specific subdomains. This could mean that the documents are fairly similar.

LDA - Evaluation

- One method to evaluate the topics is to examine the top terms. The top terms will provide an idea of what the topic is supposed to represent.
- Another method is to use the Coherence Score. The Coherence Score is used to determine the clarity of the topics.
- There are a number of different measures that can be used to determine the Coherence Score. The two most prominent measures are C_UCI and U_Mass.
- One challenge with the Coherence Score is that the different measures have different scales.
- C_UCI is based on Pointwise Mutual Information and considers the probability of co-occurence between two terms, with the reference being an external corpus.
- U_Mass is based on looking at the probability of co-occurances between top terms, with the reference being the internally used corpus.

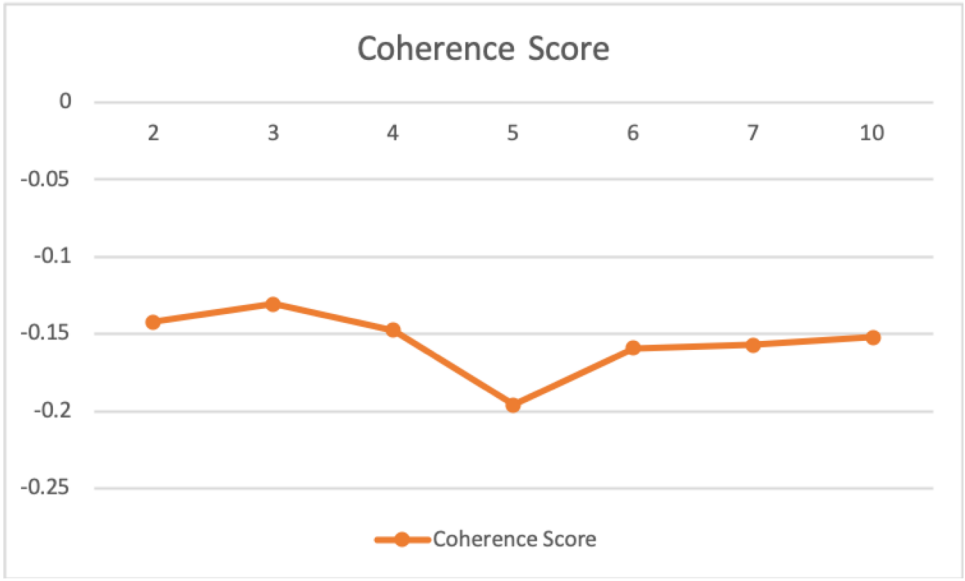
LDA - Results

- The best result across all the iterations had a coherence score of -0.13.
- It can be seen that the terms within the topics are close together and there are connections.
- An issue is there is overlap between the topics. This might indicate that there is not enough diversity amongst the topics.

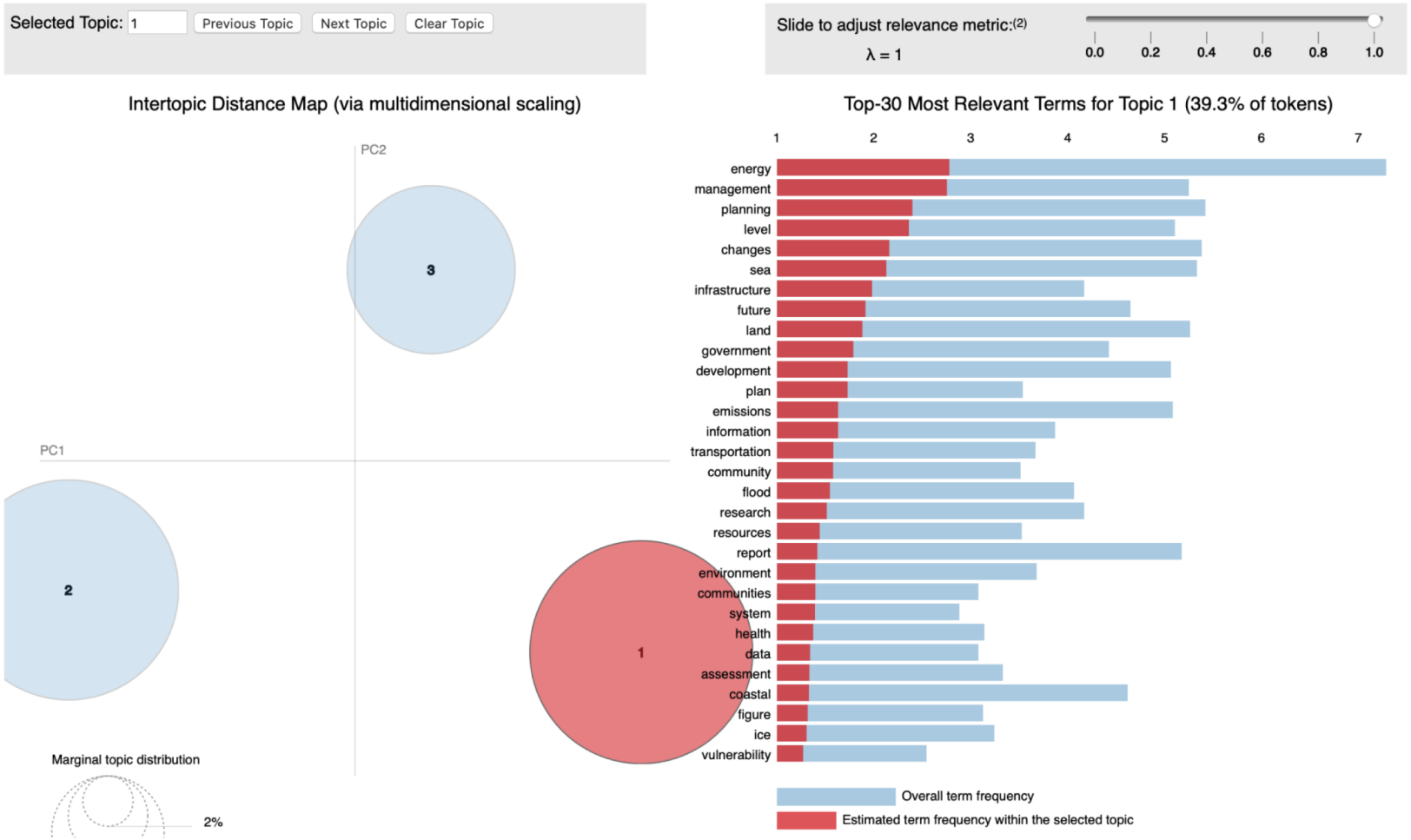
```
Number of topics: 3 Number of Terms: 10
[([(0.004631676, 'energy'),
   (0.0045930427, 'management'),
   (0.0039982614, 'planning'),
   (0.0039370125, 'level'),
   (0.003597411, 'changes'),
   (0.0035456615, 'sea'),
   (0.0033018345, 'infrastructure'),
   (0.0031888986, 'future'),
   (0.003135145, 'land'),
   (0.0029807717, 'government')],
  -0.11312160203614283),
 ([(0.005660018, 'energy'),
   (0.0044247587, 'report'),
   (0.004079819, 'emissions'),
   (0.0040084627, 'changes'),
   (0.0036828027, 'sea'),
   (0.0036512404, 'coastal'),
   (0.003472614, 'development'),
   (0.003346625, 'land'),
   (0.003226643, 'future'),
   (0.0030616391, 'research')],
  -0.12724002054886982),
 ([(0.0041694823, 'land'),
   (0.0039146347, 'planning'),
   (0.003820342, 'development'),
   (0.0035005992, 'energy'),
   (0.0034255798, 'report'),
   (0.003383334, 'coastal'),
   (0.0032530655, 'risks'),
   (0.0031254895, 'emissions'),
   (0.0030975, 'plan'),
   (0.0030801585, 'sea')],
  -0.1981268823302916)]
Random Document Distribution
[(0, 0.04762689), (1, 0.6231852), (2, 0.32918796)]
Coherence: -0.13066012318477496
```

LDA - Results

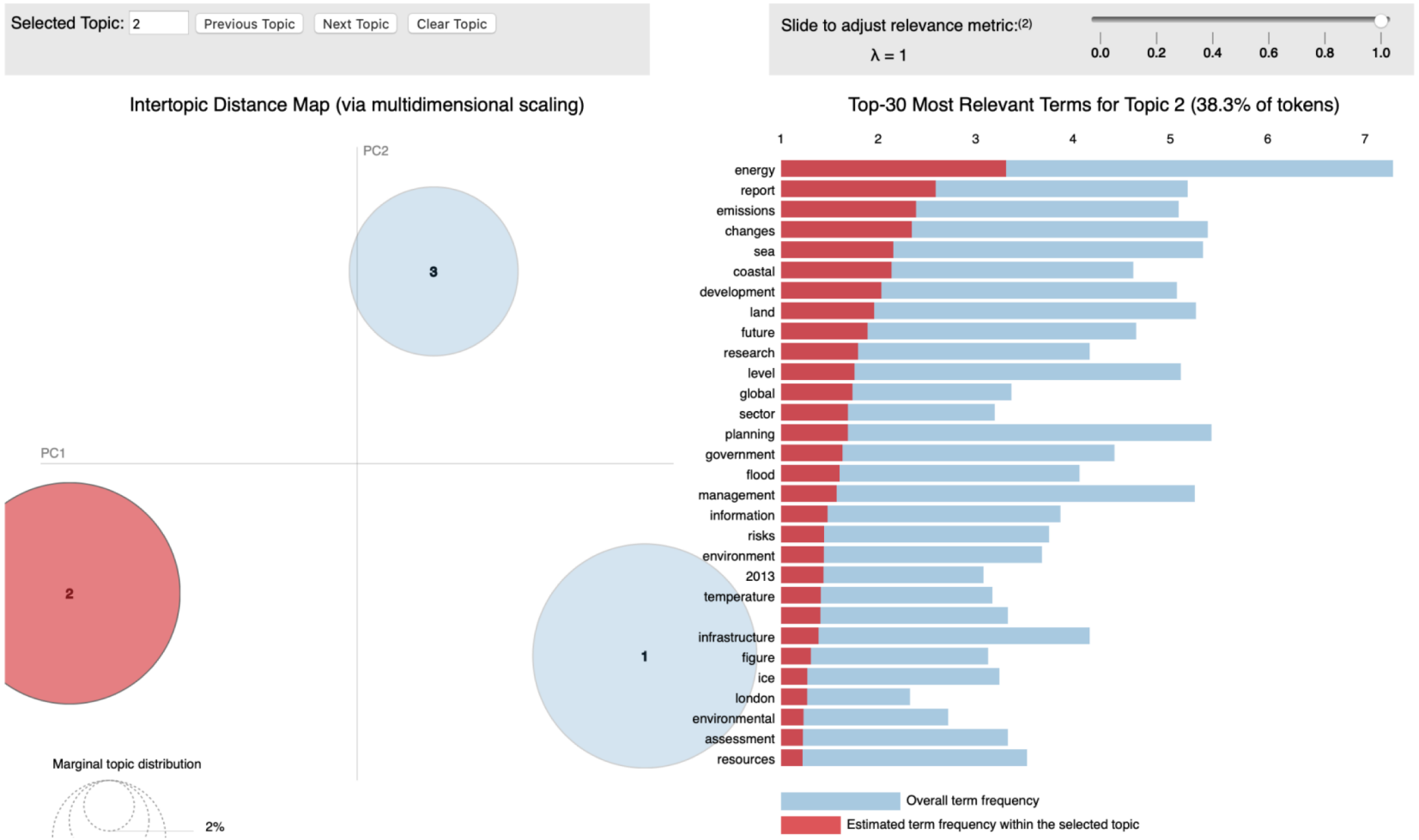
- Found that Coherence Score was better for a smaller number of topics, eventually score plateau.



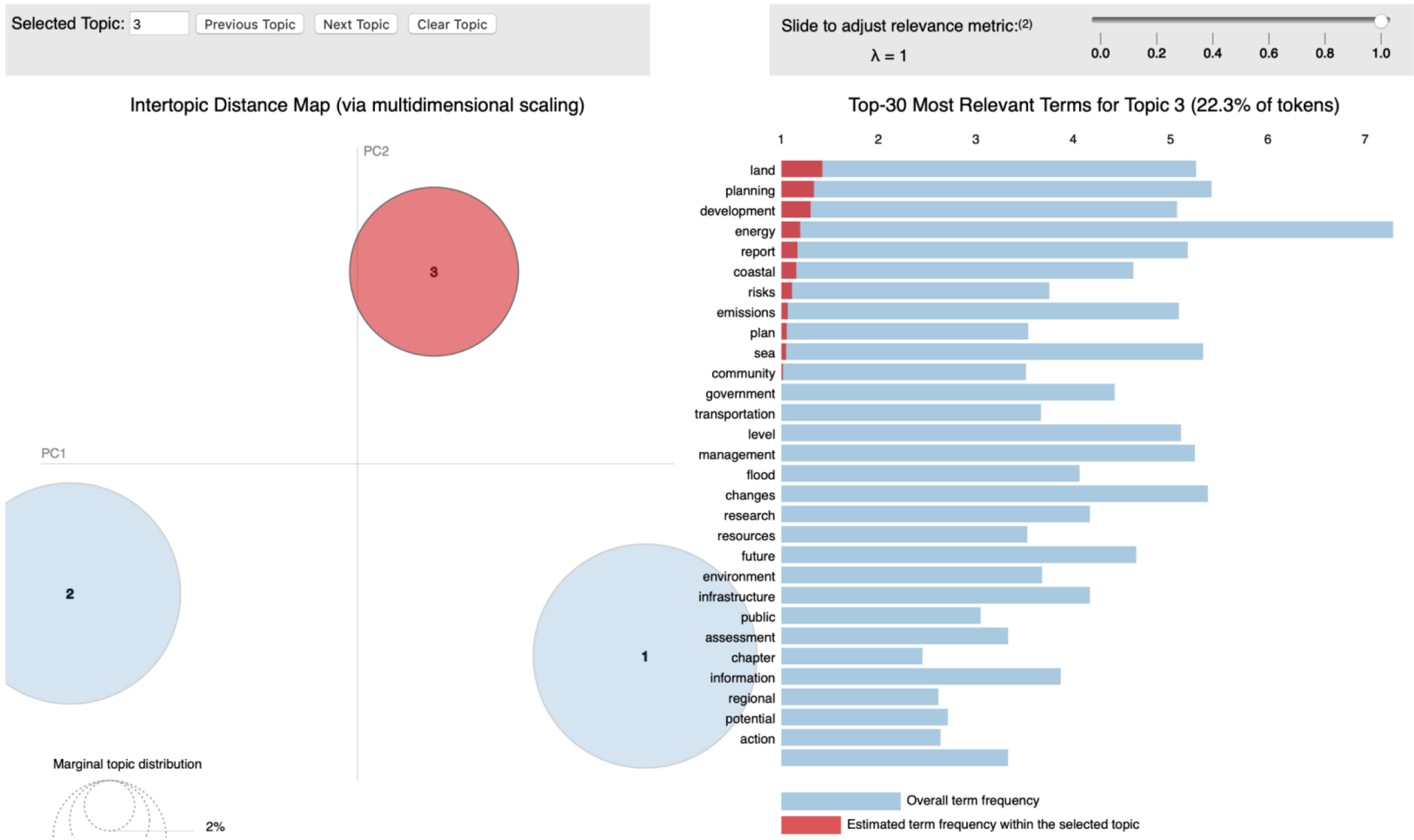
PyLDAvis - Topic 1



PyLDAvis - Topic 2



PyLDAvis - Topic 3



LDA Topic Model on Random Document

- We previously discussed about determining whether or not an unseen document belongs within the topics.
- To explore this, we utilized a pdf of one of our lectures, and fit it to the data.
- The result we got was:

```
Random Document Distribution
[(0, 0.04762689), (1, 0.6231852), (2, 0.32918796)]
```

- What the LDA model did here was assign it a probability of how it would fit within the topics that are already defined. Based off of our exploration, it does not indicate that the document does not belong.
- It does not appear that Gensim has a built-in method to perform this action with the LDA class.
- Methods of checking if the document fit include:
  - Implement heuristics in preprocessing. For example, check the name of the file of the document.
  - Examine most frequent words and compare to topic words.
  - A method that could be performed is called Doc2Vec. Here you would vectorize the entire document and then compare them to one another. The idea being that the vectorization forms of documents that are similar will be close together.

LDA Findings

- One of the challenges that was faced was that NLTK stopwords were reducing the corpus down.
- In order to stop this, a stopwords file was created and stopwords were defined based on examining the corpus tokens frequency distribution.
- Stopwords file included: common punctuation, common terms (canada, climate, change, http, www, com, and, or, etc), meaningless unicode characters (\uf0b7).
- In addition, a heuristic was applied in order to only include tokens whose length was greater than two.
- Another challenge faced was the: volume, quality, and similarity of data.
- Found that when we added new documents to our corpus, that our coherence score improved. However, this depends on the document added.

Identifying Actions
Overview

Principle Task:

- From examples of ‘actions a citizen could do’, learn to automatically identify such actions.

Our Approach:

- Look at the grammatical structure of the sentence instead of the words.
- Consider both Machine Learning and Rule-Based algorithms.

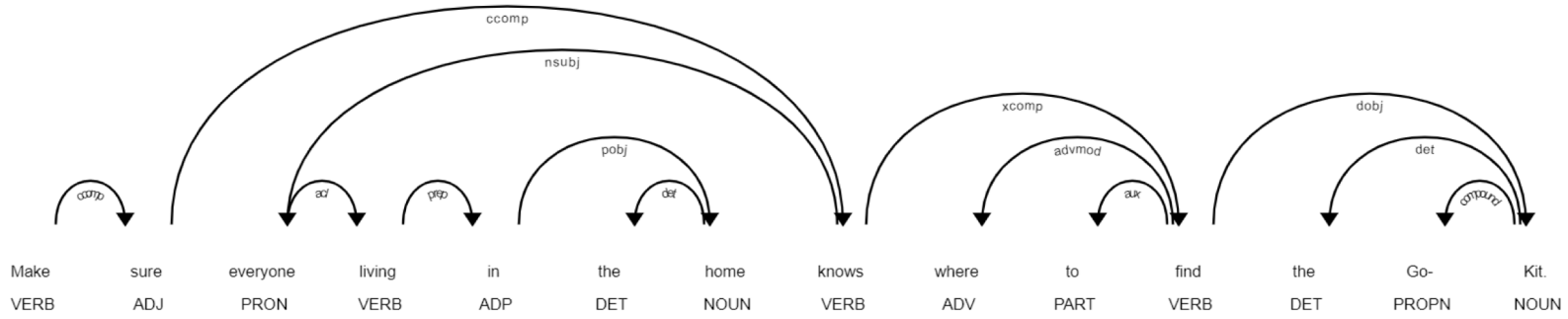
Identifying Actions
Tools

- To capture the grammatical structure of the sentence, we considered:
  - nltk POS tagging - doesn't provide enough information for our purpose
  - CLiPS modality checker - buggy
  - Stanford CoreNLP - runs in Java, so more complicated to install
  - spaCy -
    - in Python
    - easy to use
    - produces dependency parses suitable for our purpose

Identifying Actions
Text to Grammatical Tokens

1. Extract the sentences from the text file with nltk sentence_tokenizer.

2. Parse the sentences into dependency parse trees with spaCy.



3. Represent the top two levels of the trees as lists of grammatical tokens.
   `['ROOT_self_VBP', 'LEFT_nsubj_PRP', 'RIGHT_xcomp_VB', 'RIGHT_punct_.']`

4. Use the grammatical tokens instead of words in your favourite algorithm.

Identifying Actions
Logistic Regression Classifier

1. Vectorize the sequences of grammatical tokens with TF-IDF.
2. Apply varying degrees of feature reduction with TruncatedSVD.
3. Train and test Logistic Regression model with 10-fold cross-validation.
4. Typical fold scores:

	precision	recall	f1-score	support
action	0.81	0.83	0.82	83
non_action	0.81	0.79	0.80	77
accuracy			0.81	160
macro avg	0.81	0.81	0.81	160
weighted avg	0.81	0.81	0.81	160

5. Best results were obtained without Truncated SVD.
6. Retrain the model on the entire training data.
7. Test on the held-out test data.

Identifying Actions
Manual Scoring of a Small Sample

Human	No SVD	SVD-5	Sentence
action:	action:	action:	Process Infrastructure Ports, Marine & Offshore Project No.
non_action:	action:	action:	This approach will minimize the initial costs of considering SLR, and the future costs of adaptation.
action:	action:	action:	Such infrastructure should be designed and constructed to remain operational during floods.
non_action:	non_action:	action:	1.16 Sea Dike System A system of: dikes, dunes, berms or natural shorelines that provide a similar fu
non_action:	non_action:	non_action:	We are especially grateful to the Government of Canada_s Climate Change Impacts and Adaptation Prograı
non_action:	non_action:	action:	Over the years since the United Nations Framework Convention on Climate Change was first signed in Ri
non_action:	non_action:	non_action:	The voices in my head that don_t want to be seen to always be a bother, that want to be liked, that a
non_action:	non_action:	action:	The loss of future timber supply presents a very significant challenge to the communities affected be
non_action:	non_action:	action:	Long-term climate records in the region indicated a significant warming trend, particularly in winter
action:	action:	action:	Use a list of suggested adaptation options and involve the workshop articipants in identifying a list
action:	action:	action:	Assess new information as it becomes available that could provide insight into or change best practic
non_action:	non_action:	action:	(2004); Jones et al.
non_action:	non_action:	action:	The judgments can be of at least two types, and are commonly a bit of both: that is, they may be issu
non_action:	non_action:	action:	From the viewpoint of adaptation and vulnerability reduction, these consequences are very important f
non_action:	non_action:	action:	SCD promotes economic gains while also facilitating social and environmental benefits.
non_action:	non_action:	non_action:	Technological approaches to adaptation include both _hard_ technologies such as capital goods and har
action:	non_action:	action:	(European Environment Agency, 2005) For example, no-till farming with residue mulching slows soil ero
non_action:	non_action:	action:	Davidson, P.R.
non_action:	non_action:	action:	Hogg and _. Meki.
non_action:	non_action:	action:	(2005) Adaptation Policy Frameworks for Climate Change: Developing Strategies, Policies and Measures.
non_action:	non_action:	action:	Climatic Change 61: 1-8 CANADIAN COMMUNITIES_ GUIDEBOOK FOR ADAPTATION TO CLIMATE CHANGE 99 guidebook
non_action:	non_action:	action:	How climate change is understood and talked about in Yukon is grounded in scientific evidence, the Tr
non_action:	non_action:	action:	A seventeen-year study of the vegetation communities on Qikiqtaruk-Herschel Island documented rapid c
non_action:	non_action:	non_action:	There are obvious economic benefits to going forward with a FireSmart Plus approach linked to biomass
non_action:	action:	action:	This local premium may be as high as two to four- fold over imported production._ (Serecon, 2007 P. A
non_action:	action:	action:	Mayo Region Climate Change Adaptaton Plan.
non_action:	non_action:	action:	Retrieved from: http://www.yukon-news.com/ RESEARCH NORTHWEST & MORRISON HERSHFIELD 40 Yukon _State o
non_action:	non_action:	action:	The project is part of the wider initiative, the BC Regional Adaptation Collaborative (RAC) to help c
action:	action:	action:	Planning can be initiated either through voluntary means or by regulation.
non_action:	non_action:	action:	Changes in air temperature and precipitation patterns are noticeably affecting our weather, water cyc
action:	non_action:	action:	_ Restore riparian and instream habitat.

Identifying Actions
Results on Held-Out Test Data

- Logistic Regression without Feature Reduction

Metric	Score
Accuracy	0.71
Precision	0.67
Recall	0.63
F1	0.67

- Logistic Regression with 5 Features

Metric	Score
Accuracy	0.35
Precision	0.26
Recall	1.0
F1	0.41

- In general, the more we reduced the number of features, the more the algorithm labelled everything an ‘action’.

Identifying Actions
Rule-Based Solution

- Obvious pattern visible in frequency counts of grammatical tokens for actions

```
[('ROOT_self_VB RIGHT_dobj_NN RIGHT_punct_.', 47),
 ('ROOT_self_VB RIGHT_dobj_NNS RIGHT_punct_.', 28),
 ('ROOT_self_VB RIGHT_dobj_NN RIGHT_prep_IN RIGHT_punct_.', 18),
 ('ROOT_self_VB RIGHT_acomp_JJ RIGHT_punct_.', 17),
 ('ROOT_self_VB RIGHT_prep_IN RIGHT_punct_.', 17),
 ('ROOT_self_VB RIGHT_dobj_NN RIGHT_cc_CC RIGHT_conj_VB RIGHT_punct_.',
 ('ROOT_self_VB RIGHT_dobj_NN RIGHT_advcl_VB RIGHT_punct_.', 9),
 ('ROOT_self_VB RIGHT_cc_CC RIGHT_conj_VB RIGHT_punct_.', 9),
```

- Created simple hard-coded rule

```
if candidates[i][0] == 'ROOT_self_VB' and candidates[i][1].startswith('RIGHT_dobj_NN'):
```

- Results on held-out test data

Metric	Score
Accuracy	0.85
Precision	1.0
Recall	0.77
F1	0.87

- For this specific task, a one-line rule was the best performer.

Challenges

- Data Variability
  - PDF documents were not in consistent format.
  - PDF extraction and preprocessing is labour intensive.
  - PDF extraction tools themselves are fairly easy to use, dealing with the output and wrangling it to a useable format can be difficult and imprecise.
- Amount of data
  - In order to deal with a smaller dataset, time was spent in order to enrich it.
- Monolithic data
  - Many documents were large and covered many topics. This made it more difficult to distinguish individual topics from each other.
- Human Scoring Bias
  - It was difficult to remain objective in the manual scoring of the test results at the end, after having worked with the data and models for a long time.

Lesson Learnt

- Better to work with a larger dataset.
- PDF extraction was a valuable exercise. PDFs can be difficult to work with and the results from the different tools will vary.
- It is better to use fewer topics. Noticed that fewer topics had a better Coherence Score, eventually the score plateaus.
- LDA is not a very good tool for small datasets.
- Parsing sentences into grammatical structure can yield good results for some problems.
- Sometimes a hard-coded rule can be the right tool for the job.

Next Steps

- If we were to continue this project further:
  - Work on acquiring more data and observe impact on results.
  - Try other binary classifiers and compare results.
  - Test Doc2Vec and other solutions for identifying whether a document fits with the Climate Change theme and belongs within the Corpus.
  - Identify possible methods to leverage the metadata associated with the PDF documents.
  - Consider how these NLP solutions would fit into your workflow and system. What areas may require a Human Touchpoint?