Designing Faceted Recommendation System for Scientific Articles

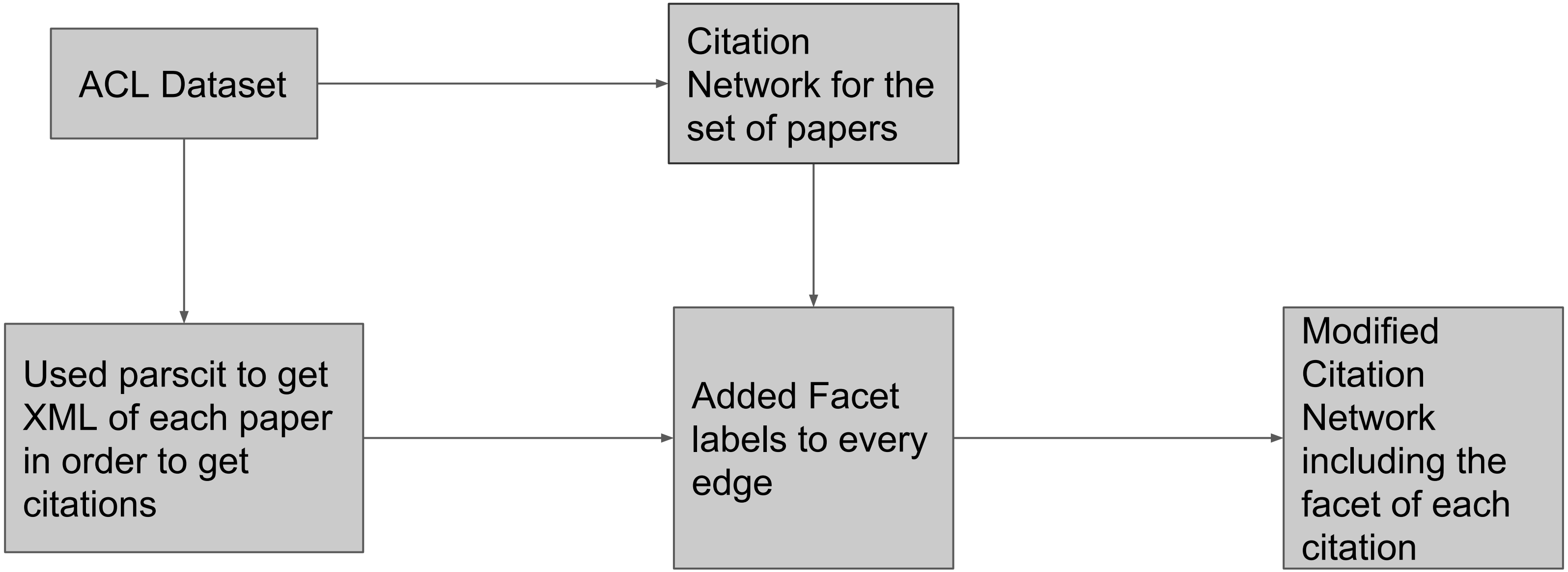
Mentor - Sandeepan Sikdar

Kalpit Chittora
Beena Mahato
Ankit Pandey
Aditi Garg
Amul Patwa

Introduction

- A smart recommendation engine should be able to organize the recommended papers into multiple facets/tags such as background, alternative approaches, methods and comparison.
- We have used the AAN dataset which is an assemblage of all papers included in ACL2 publication venue and categorize the citation links based on their occurrence in various sections of the paper.

Model (Citation Network)



Model (Random Walk)

<u>Top 100 papers</u>
(from similarity)

<u>All papers within 2-hop distance</u>
(from citation network)

<u>Induced Graph</u>
(edge_label = Facets)
(edge_weights=*Similarity*)

Random Walker Restart Algorithm (using bidirectional transportation)

Probability of recommending a paper

<u>Baseline:</u> *Similarity* = Cosine_similarity

<u>Improvised:</u> *Similarity* = (1- JSDivergence)

Creating the Citation Network

Xml generated from parscit


```
-<booktitle>
   Representativeness in corpus design. Literary and Lin
 </booktitle>
 <pages>8--4</pages>
-<contexts>
 -<context position="2047" citStr="Biber, 1993" sta
     ystems. Exchanging experiences and developing gui
     development of computational models of speech, la
     field of natural language dialogue systems with that
     sampling size, representativeness in corpus design a
     Crowdy, 1993; Biber, 1993)). Also the neighboring a
     to have advanced further. Some work have been do
     (Dahlback et al., 1998), on measures for inter-rater
```


Section-Facet mapping file

25464	Unsuperv ised Word Sense.	M
25465	Linking Sense to Translation.	M
25466	Writing Systems.	M
25467	Introduction: Corresponding Entities.	I
25468	The Model: Term Subset Coupling by.	M
25469	Algorithms: Balancing Within-Cluster.	M
25470	Results: Hierarchy and Granularity.	RE
25471	Structure of the IE System.	M
25472	General and Specific Pat-.	M
25473	Example-based Acquisition.	M

To generate the citation network, the “Section” strings were compared to entries in “Section-facet_mapping” file to obtain the facet.

Citation Network

- The citation network is a dictionary containing citation entries (along with the facet labels) corresponding to every paper.

```
In [15]: outcite_2

Out[15]: {'P03-1014': {'C': [],
           'E': [],
           'I': [],
           'M': [],
           'MM': [],
           'None': [],
           'RE': ['E03-1052'],
           'RW': []},
          'D08-1091': {'C': [],
           'E': [],
           'I': ['A00-2018',
            'H05-1064',
            'P01-1042',
            'P04-1013',
            'P05-1022',
            'P06-1055',
            'P08-1067',
            'P08-1109',
            'W04-3201'],
```

Tf-Idf Vectorizer

- Generated vectors corresponding to every document using Tf-Idf vectorizer.
- We have used the stop words of general english language.
- Removed words occurring in less than 1% of the papers.
- The Vectorizer also performs add-1 smoothing.

```
In [23]: print(matrix)

  (0, 6540)	0.0201032005969
  (0, 5486)	0.0104051898155
  (0, 444)	0.156021584466
  (0, 1207)	0.304374419524
  (0, 8649)	0.240782954318
  (0, 6800)	0.0182487880549
  (0, 8043)	0.0162286058078
  (0, 5581)	0.0204536566907
  (0, 2541)	0.0148207916092
  (0, 1846)	0.0312596963943
  (0, 4894)	0.0507675349899
  (0, 8817)	0.0129132282523
  (0, 10624)	0.0201647956815
  (0, 6732)	0.0662848983052
  (0, 6497)	0.0240131088803
  (0, 31)	0.0048647662426
  (0, 7113)	0.00483935864626
  (0, 2571)	0.00861545124645
  (0, 6541)	0.124742174975
  (0, 880)	0.022177162555
  (0, 3496)	0.0128653557644
  (0, 7918)	0.081385121958
  (0, 10354)	0.0430177614228
  (0, 801)	0.0123018048067
  (0, 5047)	0.0705285837728
```

Cosine Similarity

- We implemented cosine similarity using the tf-idf matrix.
- As a result we got the probabilistic similarity between two documents which we further use to generate the rank of recommended papers for a query paper.
- The output of this matrix was used as weights of the edges of induced graph.

Cosine Similarity

```
In [4]: print(cos_sim_array)

        [[ 1.          0.11282674  0.10422889 ...,  0.06912642  0.06474793
           0.07643561]
         [ 0.11282674  1.          0.1930469  ...,  0.05117951  0.06028679
           0.05549579]
         [ 0.10422889  0.1930469   1.         ...,  0.06145951  0.06094063
           0.24324046]
         ...,
         [ 0.06912642  0.05117951  0.06145951 ...,  1.          0.58039218
           0.09067514]
         [ 0.06474793  0.06028679  0.06094063 ...,  0.58039218  1.          0.07659732]
         [ 0.07643561  0.05549579  0.24324046 ...,  0.09067514  0.07659732  1.        ]]

In [5]: get_graph_txt('A00-1005', outcite 2, incite 2)
```

Snapshot of Cosine Similarity Matrix

Latent Dirichlet Allocation

- Topic Modeling using Latent Dirichlet Allocation(LDA).
- Num_topics = 100

```
.....check if lda_model exists.....

 .....printing LDA model.....

(92, u'0.013*"discours" + 0.009*"relat" + 0.007*"state" + 0.006*"annot" + 0.006*"semant"')
(24, u'0.012*"segment" + 0.010*"train" + 0.009*"charact" + 0.006*"featur" + 0.005*"english"')
(61, u'0.008*"transliter" + 0.007*"train" + 0.006*"pair" + 0.006*"featur" + 0.005*"text"')
(54, u'0.012*"translat" + 0.007*"sentenc" + 0.007*"phrase" + 0.007*"tree" + 0.006*"featur"')
(47, u'0.007*"algorithm" + 0.006*"form" + 0.006*"rule" + 0.005*"class" + 0.005*"tree"')
(65, u'0.011*"sentenc" + 0.007*"translat" + 0.007*"relat" + 0.006*"featur" + 0.006*"phrase"')
(91, u'0.026*"parser" + 0.022*"depend" + 0.021*"pars" + 0.007*"featur" + 0.007*"train"')
(22, u'0.015*"featur" + 0.011*"relat" + 0.008*"train" + 0.007*"label" + 0.006*"depend"')
(10, u'0.012*"learn" + 0.009*"featur" + 0.008*"train" + 0.008*"tree" + 0.007*"sentenc"')
(71, u'0.012*"train" + 0.007*"sentenc" + 0.006*"score" + 0.005*"learn" + 0.005*"pars"')
```

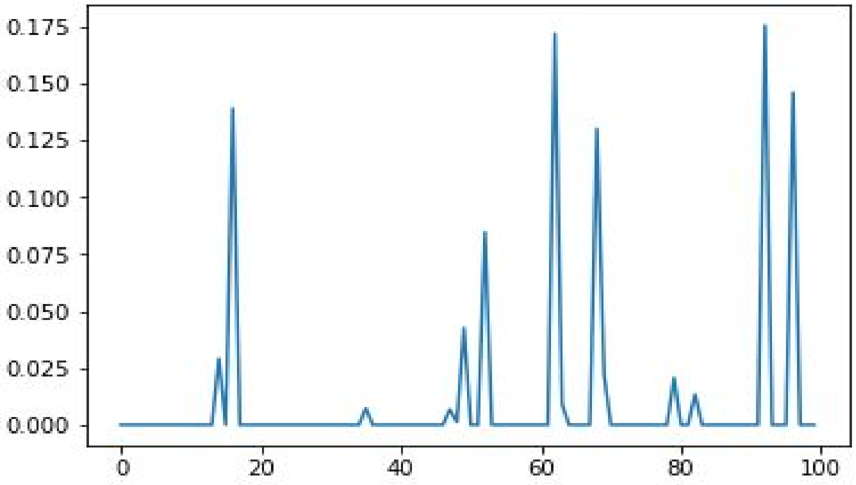
```
[13]:
# In[16]:
```

Latent Dirichlet Allocation

- Topic-Probability Distribution for ‘A00-1009’

```
plt.plot(get_doc_topics(ldamodel, corpus[paper_array.index('A00-1009')]))
```

```
Out[42]: [<matplotlib.lines.Line2D at 0x7f7fe3d50950>]
```



Jensen–Shannon Divergence

1. Method for measuring the similarity between two probability distributions.

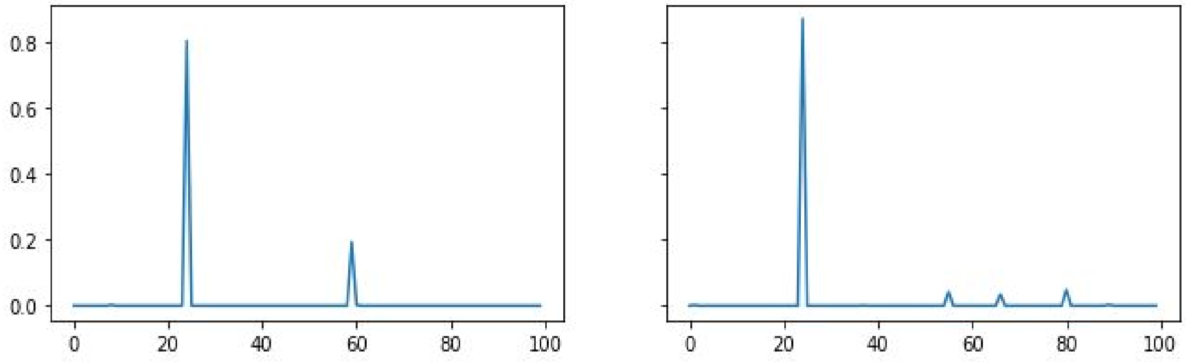
$$JSD(P \parallel Q) = \frac{1}{2}D(P \parallel M) + \frac{1}{2}D(Q \parallel M)$$

$$M = \frac{1}{2}(P + Q)$$

Low Divergence = High Similarity

```
In [20]: plt.rcParams['figure.figsize'] = (11, 3)
         f, (ax1, ax2) = plt.subplots(1, 2, sharey=True)
         ax1.plot(get_doc_topics(ldamodel, corpus[paper_array.index('N09-1007')]))
         ax2.plot(get_doc_topics(ldamodel, corpus[paper_array.index('P06-2056')]))

Out[20]: [<matplotlib.lines.Line2D at 0x7fb4e5b84990>]
```

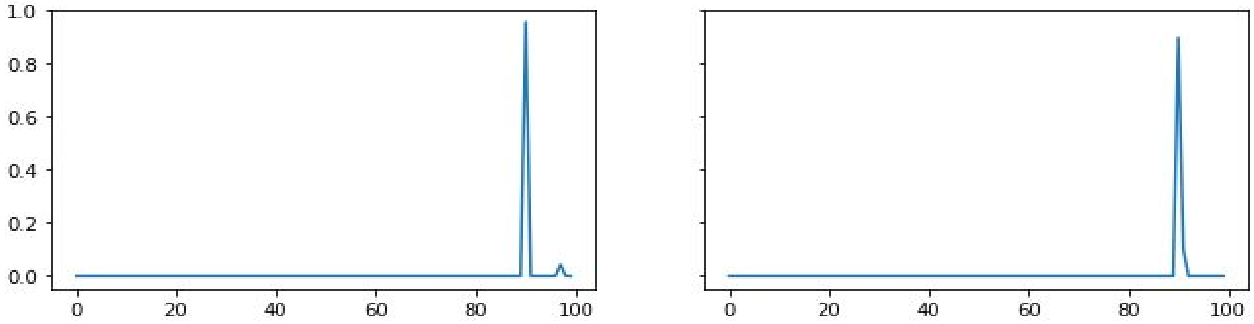


P06-2056 Unsupervised Segmentation Of Chinese Text By Use Of Branching Entropy
N09-1007 A Discriminative Latent Variable Chinese Segmenter with Hybrid Word/Character Information
JSDiv = 0.11

Low Divergence = High Similarity

In [24]:
```
plt.rcParams['figure.figsize'] = (11, 3)
f, (ax1, ax2) = plt.subplots(1, 2, sharey=True)
ax1.plot(get_doc_topics(ldamodel, corpus[paper_array.index('N04-1030')]))
ax2.plot(get_doc_topics(ldamodel, corpus[paper_array.index('W05-0639')]))
```

Out[24]:
```
[<matplotlib.lines.Line2D at 0x7fb4e5592c50>]
```

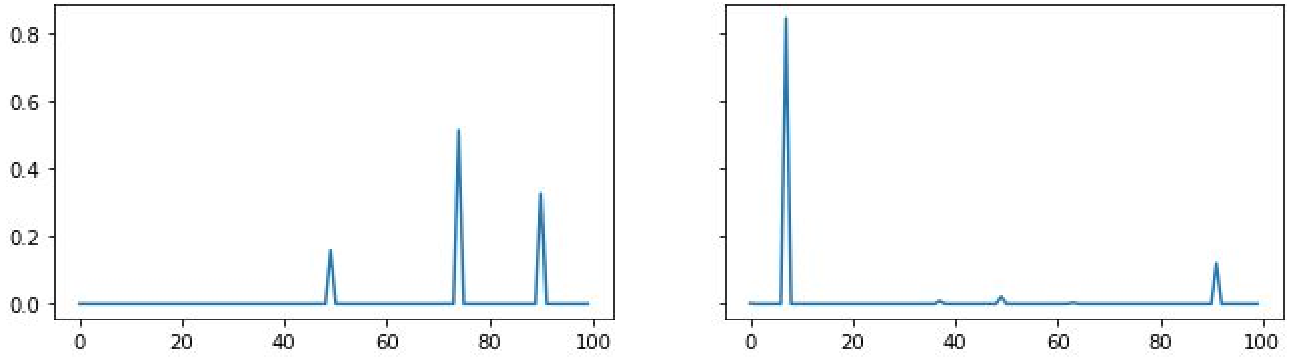


N04-1030 Shallow Semantic Parsing Using Support Vector Machines
W05-0639 The Integration Of Syntactic Parsing And Semantic Role Labeling
JSDiv = 0.05

High Divergence = Low Similarity

```
In [23]: plt.rcParams['figure.figsize'] = (11, 3)
         f, (ax1, ax2) = plt.subplots(1, 2, sharey=True)
         ax1.plot(get_doc_topics(ldamodel, corpus[paper_array.index('C04-1100')]))
         ax2.plot(get_doc_topics(ldamodel, corpus[paper_array.index('P06-2004')]))

Out[23]: [<matplotlib.lines.Line2D at 0x7fb4e573f250>]
```



C04-1100 Question Answering Based On Semantic Structures

P06-2004 The Effect Of Corpus Size In Combining Supervised And Unsupervised Training For Disambiguation

JSDiv = 0.66

Random Walk with Restarts

- Induced Citation Graph: From the citation network, a directed induced graph corresponding to the query paper is generated.

- Baseline: (**hop-limit = 2** or **cosine_similarity > 0.25**)
  Improvised: (**hop-limit = 2** or **JSDiv < 0.51**)

- Weighted edges: Each edge is assigned a probability based on the (**similarity) or (1-divergence)** of corresponding nodes(papers).

Random Walk with Restarts

$$p^{t+1} = (1-r)\, A\, p^{t} + p^{0}$$

A: Transportation Matrix
P^t: probability matrix at time = t
r : restart probability = 0.4

- The relevance score of node *j* with respect to node *i* is defined by the steady-state probability $r_{i,j}$ that the walker will finally stay at node *j*.

- “Code developed by Zhang H, Schaefer M, Crawford J, Kiel C, Serrano L, and Cowen LJ”

Results

A00-2004 (Citations for Method Facet)
Advances In Domain Independent Linear Text Segmentation

Cosine Similarity

J06-3003
Similarity of Semantic Relations

C10-1142
Estimating Linear Models for Compositional Distributional Semantics

P93-1001
Char Align: A Program For Aligning Parallel Texts At The Character Level

N01-1027
Identifying User Corrections Automatically In Spoken Dialogue Systems

Jensen Shannon Divergence

P93-1001
Char Align: A Program For Aligning Parallel Texts At The Character Level

W01-0514
Latent Semantic Analysis For Text Segmentation

A94-1013
Adaptive Sentence Boundary Disambiguation

A97-1004
A Maximum Entropy Approach To Identifying Sentence Boundaries

Results

Surveying with 4 people in our group, **JSDivergence** performed better than **Cosine_similarity** 60% of the times.

Name	Subject1	Subject2	Subject3	Subject4
Set-1 (**Cosine Similarity**)	4	5	4	3
Set-2 (**JSDivergence**)	6	5	6	7

Survey performed on 10 sample papers.