COL764: Assignment 2

Archisman Biswas, 2021MT10254

25 September 2024

Contents

1 Relevance Model for Retrieval **2**
1.1 Model implementation details 2
1.2 Text preprocessing 2

2 Query Expansion using Local Embeddings **3**
2.1 Training 3
2.2 Query Expansion 3

3 Query Expansion using Generic Embeddings **4**

4 Results **4**
4.1 Local Word2Vec embedding 4
4.2 Generic Word2Vec/GloVe emdedding 6
4.3 Comparison across models 7

Note: All measures are given w.r.t programs run on Apple Silicon M2 chip

1 Relevance Model for Retrieval

In this part, the Query-Translation model for re-ranking documents is implemented.
We estimate $P(q|R = 1, d)$, where q, d is query given and current relevant document respectively. Dirichlet's Smoothing [Chen & Goodman, 1998] states that:

$$\hat{P}(t|M) = \frac{f_{t,d} + \mu \hat{P}_C(t)}{|d_j| + \mu}$$

$$\log(P(q|d_j)) = \sum_{i=1}^{n} \log \left(\frac{f_{t_i,d} + \mu \hat{P}_C(t_i)}{|d_j| + \mu} \right)$$

Here $\mu = 1000$, as per tuned results from 3

The underlying distribution given a document is assumed as multinomial here and the unigram priors are learnt. In the dataset, **24** given queries retrieve **100** documents each. This makes **2400** documents in all, which are used to train the vocabulary and compute $\hat{P}_C(t)$. This makes the collection (in the above equations) a larger set of documents spanning across topics, as it is supposed to be. The file containing all documents is used only to fetch the actual content of these 2400 documents.

> This set of 2400 documents would be referred to as **corpus** from here onwards.

1.1 Model implementation details

These are the functions w.r.t to the Query Translation model which are used later by other parts:

- `train_query_translation_model(query_path, top100_path, coll_path, model_path)`
  The LM is trained here and the parameters, like vocabulary words and their corresponding frequencies in the are stored in `model_path` in the form of a `key:value` dictionary as `'models/qt_model'`. The other variables have their usual meanings.

- `get_reranked_results(query_docs_dict, required_docs_dict, lm_path)`
  This function is used to re-rank, after the LM is trained. The location to the trained LM is `lm_path`, while `query_docs_dict` and `required_docs_dict` stores the retrieval details per query and corpus details respectively.

1.2 Text preprocessing

The vocabulary above is trained on the entire corpus text.
The delimiters used for tokenization are `['.', ' ', ':', ';', '"', "'", '.', '?', '!', ',', '\n', '~', '‘', '(', ')', '/', '#', '*', '%', '+', '-', '[', ']', '{', '}', '@', '^', '<', '>']`
For tokenization, only English alphabets were considered. Although various types of characters were experimented with, using only English alphabets proved to be the most efficient in terms of both accuracy and time.
The size of vocabulary is **149989**
During query expansion in 2 and 3, stopword removal is done on the document text when the respective embedding models are trained and the retrived documents are re-ranked for each query. This step shows a good improvement on the retrieval efficiency.

2 Query Expansion using Local Embeddings

In this part, Word2Vec (Skipgram) is implemented. For each query, a model is trained on the set of documents which were retrieved. These documents form the pseudo-relevance set, on which we train our Word2Vec model and then expand the query. This expanded query is then sent to `get_reranked_results` to get the reranked results. The final results and evaluations are shown in 4.1

2.1 Training

Hyperparameter	**Value**
Embedding Dimension	100
Negative Sampling Rate	5
Context Size	5
Minimum Count	2
Step Size	0.001
Epochs	50
Patience	5

Table 1: Model Hyperparameters

All variables have usual meanings. Patience is the number of epochs to wait if there is no change in loss. After training is completed, the models along with the vocabulary of words learnt are saved to `'./models/local'` and `'./models/vocab'` respectively.

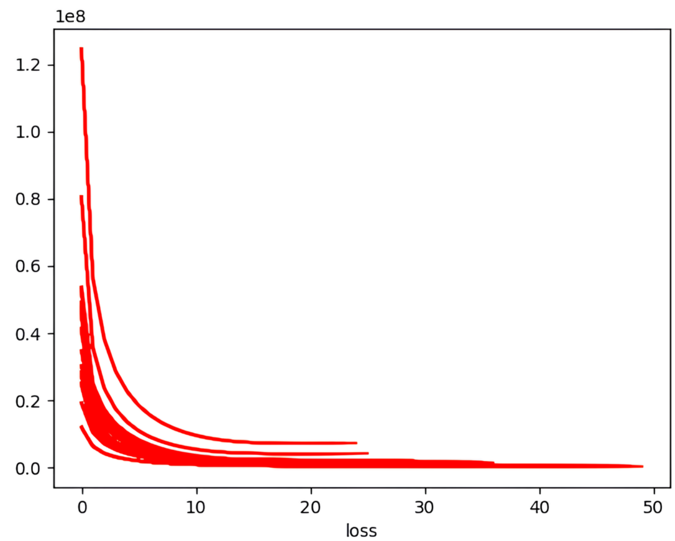

Figure 1: Training loss for queries after each epoch

2.2 Query Expansion

[Diaz et al., 2016] During local training, we should do it on query-specific set of topical documents. This leads us to assuming that the retrieved documents per query should be our pseudo-relevance set. The following leads us to compute some top n terms for query expansion:
If $|V|$ is the vocabulary size as learnt by the w2v model, let $\mathbf{U}$ be an $|V| \times k$ term embedding matrix. If $\mathbf{q}$ is a $|V| \times 1$ column term vector for a query (after hot-encoding the query words), then the expansion term weights are $\mathbf{U}\mathbf{U}^T\mathbf{q}$. We then take the words corresponding to the top n values of this vector.
Number of expansion words for each query is **10**

3 Query Expansion using Generic Embeddings

This part is similar to 2 except that pre-trained generic **Word2Vec** and **GloVe** embeddings are used here. Embedding dimension is **300**

Model	Vocabulary size
Word2Vec	302868
GloVe	400001

Table 2: Intermediate vocabulary sizes of each model

Number of expansions is **10**. And query expansion is done as it is mentioned in 2.2. The final results and evaluations are shown in 4.2

4 Results

For evaluation of the results, `nDCG@{5, 10, 50}` will be used

$$DCG[k] = \sum_{i=1}^{k} \frac{G[i]}{\log_2(i+1)}$$

$$nDCG[k] = \frac{DCG[k]}{DCG^{ideal}[k]}$$

where $G[i]$ is the relevancy of the i'th retrieved document and $DCG^{ideal}[k]$ is on the ideal vector formed on sorting the documents w.r.t relevancy in non-increasing order.

4.1 Local Word2Vec embedding

Query ID	Initial Retrieval	Local w2v
42255	0.0	0.4396592698888862
47210	0.3835663673713356	0.7442890884191096
67316	0.3166798143039299	0.09691603875641078
135802	0.1400365039073521	0.5993061131703965
156498	0.0	0.14652668670302232
169208	0.04868944994756881	0.353089811476028
174463	0.0	0.0
258062	0.16958010263680806	0.21398626473452756
324585	0.38427838394797204	0.0
330975	0.0	0.27082852073795854
332593	0.2285839774074794	0.18773177839127778
336901	0.0	0.0
673670	0.0	0.0
701453	0.0	0.1159310140249899
730539	0.0	0.0
768208	0.0	0.6877623875401735
877809	0.1785236825433434	0.6191463544777535
911232	0.0	0.40413261305110937
938400	0.0	0.0
940547	0.0	0.0
997622	0.04373502583744726	0.40378964041468124
1030303	0.0	0.2807721888661444
1037496	0.1027791229791241	0.28307289892484966
1043135	0.2730353674546231	0.281663127986316

Table 3: `nDCG[5]` values for Local w2v

Query ID	Initial Retrieval	Local w2v
42255	0.0	0.4396592698888862
47210	0.3209613490677604	0.6199890360954773
67316	0.2900213566936929	0.07784291905153429
135802	0.11446761440147785	0.5688104764183949
156498	0.04195838781243983	0.3889101215735702
169208	0.03159612145651692	0.3648165729859299
174463	0.0	0.0
258062	0.12222609441075938	0.23134823887974634
324585	0.3669164148705767	0.0
330975	0.0	0.21002787876934015
332593	0.24353212324541304	0.2859356720798296
336901	0.0	0.0
673670	0.0	0.0
701453	0.0	0.22210098254131785
730539	0.07422802918647313	0.04013443988134614
768208	0.0	0.68204179669054
877809	0.23691051308354	0.615922796524412
911232	0.0	0.38316709808686505
938400	0.0	0.1427951440361373
940547	0.0	0.0
997622	0.09912316120484518	0.35660269895598956
1030303	0.0	0.4807983320944877
1037496	0.11211657539138528	0.39271393296058044
1043135	0.4440916044944476	0.3609687380931054

Table 4: `nDCG[10]` values for Local w2v

Query ID	Initial Retrieval	Local w2v
42255	0.0	0.5572164479984186
47210	0.48291922948202565	0.6423539152736695
67316	0.4116177295993815	0.28469140473588483
135802	0.23174592060046845	0.7491362592797363
156498	0.23696510012417846	0.4563453996605261
169208	0.3547518339036082	0.502877945334369
174463	0.0	0.13954241625369654
258062	0.3680564732576316	0.36284927547911644
324585	0.3669164148705767	0.10282962827291525
330975	0.2797074865997598	0.4915883091105434
332593	0.3754134198948221	0.4654074483556922
336901	0.0	0.2190205378823039
673670	0.0	0.0
701453	0.23753765850238348	0.4836551334416782
730539	0.3194001165672848	0.25114029296212764
768208	0.15850437176150103	0.8339040819101247
877809	0.4140961040371294	0.6231581011469058
911232	0.0	0.5461591624577695
938400	0.25238558479513584	0.3892015074384469
940547	0.23052118288074838	0.2969072703708546
997622	0.269750172476022	0.5669126192173577
1030303	0.0	0.6240823208785283
1037496	0.3949829483909051	0.4730673601309506
1043135	0.45370446161579964	0.4219886908099841

Table 5: `nDCG[50]` values for Local w2v

4.2 Generic Word2Vec/GloVe emdedding

Query ID	Initial Retrieval	Generic Word2Vec	Generic GloVe
42255	0.0	0.0	0.18935094108080652
47210	0.3835663673713356	0.6321081344913987	0.6534081967245269
67316	0.3166798143039299	0.0	0.11884759419488326
135802	0.1400365039073521	0.10834702930374003	0.36842620471597837
156498	0.0	0.18241828703345414	0.3584621982302271
169208	0.04868944994756881	0.20052345343276654	0.569287766771529
174463	0.0	0.0	0.0
258062	0.16958010263680806	0.16958010263680806	0.4239502565920201
324585	0.38427838394797204	0.13284008937902195	0.28700877384770807
330975	0.0	0.0	0.18148604814387062
332593	0.2285839774074794	0.2570501543117026	0.3609270624073952
336901	0.0	0.0	0.0
673670	0.0	0.0	0.0
701453	0.0	0.2714155130749646	0.14087583058284325
730539	0.0	0.1729513136981354	0.0
768208	0.0	0.4625280318545906	0.32703966649239957
877809	0.1785236825433434	0.21836974929469322	0.5498075465817568
911232	0.0	0.3840207178683732	0.30721657429469856
938400	0.0	0.0	0.09737889989513762
940547	0.0	0.1785236825433434	0.0
997622	0.04373502583744726	0.6399453854227659	0.49125969208957576
1030303	0.0	0.2807721888661444	0.0
1037496	0.1027791229791241	0.2588334835033088	0.28307289892484966
1043135	0.2730353674546231	0.07616128966473121	0.3578244176510472

Table 6: `nDCG[5]` values for generic w2v

Query ID	Initial Retrieval	Generic Word2Vec	Generic GloVe
42255	0.0	0.0	0.3359040310437686
47210	0.3209613490677604	0.5733123941201149	0.5489548612651134
67316	0.2900213566936929	0.0	0.18053253914170223
135802	0.11446761440147785	0.12672010740446477	0.45656737444792733
156498	0.04195838781243983	0.23478316099620303	0.370373018736761
169208	0.03159612145651692	0.2574335509053561	0.5719827521984512
174463	0.0	0.0	0.0
258062	0.12222609441075938	0.2422681971645567	0.45771182847212455
324585	0.3669164148705767	0.3424064461494894	0.47680853983449406
330975	0.0	0.09382844813642036	0.22211018439732447
332593	0.24353212324541304	0.2786530423128658	0.3036481891247039
336901	0.0	0.09248140402783628	0.0
673670	0.0	0.0	0.0
701453	0.0	0.4517437755002516	0.3719111974394684
730539	0.07422802918647313	0.2174279791038344	0.0759660367921986
768208	0.0	0.514569705190811	0.3693924025031954
877809	0.23691051308354	0.16195671984435944	0.45458779592229565
911232	0.0	0.4559868108835903	0.4275089976945198
938400	0.0	0.0	0.06319224291303384
940547	0.0	0.22603635656736332	0.0
997622	0.09912316120484518	0.538907517864472	0.36938132291073805
1030303	0.0	0.4807983320944877	0.0
1037496	0.11211657539138528	0.27399670329733267	0.3918138956089032
1043135	0.4440916044944476	0.1551655869282512	0.4579260457894912

Table 7: `nDCG[10]` values for generic w2v

Query ID	Initial Retrieval	Generic Word2Vec	Generic GloVe
42255	0.0	0.13764271478152218	0.36200566301261644
47210	0.48291922948202565	0.5818538243009825	0.6672052183584957
67316	0.4116177295993815	0.2667640341539251	0.4054843548739297
135802	0.23174592060046845	0.345659415906705	0.5731678116778094
156498	0.23696510012417846	0.3938380712168005	0.6608647317151876
169208	0.3547518339036082	0.4887069676663055	0.5789226653388191
174463	0.0	0.15666060496608944	0.10885343689027521
258062	0.3680564732576316	0.30136413144147867	0.42415852790152186
324585	0.3669164148705767	0.3424064461494894	0.47680853983449406
330975	0.2797074865997598	0.41882385943629	0.4150947358563668
332593	0.3754134198948221	0.4437744864720997	0.49240999972463523
336901	0.0	0.23081165151591815	0.3320483771538352
673670	0.0	0.0	0.0
701453	0.23753765850238348	0.5469551477588891	0.5144255592153518
730539	0.3194001165672848	0.45713168021296097	0.18848415024929327
768208	0.15850437176150103	0.59858423444186	0.5473779259400818
877809	0.4140961040371294	0.3330362544087127	0.5039733066571125
911232	0.0	0.626128353732397	0.6016768423615608
938400	0.25238558479513584	0.32667252135185076	0.38616290877814735
940547	0.23052118288074838	0.49914828097235475	0.20688419099008734
997622	0.269750172476022	0.6815952490612387	0.5898666170134278
1030303	0.0	0.6199968981339178	0.2443993286653967
1037496	0.3949829483909051	0.5226821566527452	0.6139705867664907
1043135	0.45370446161579964	0.43604143487921776	0.5584428967678272

Table 8: `nDCG[50]` values for generic w2v

4.3 Comparison across models

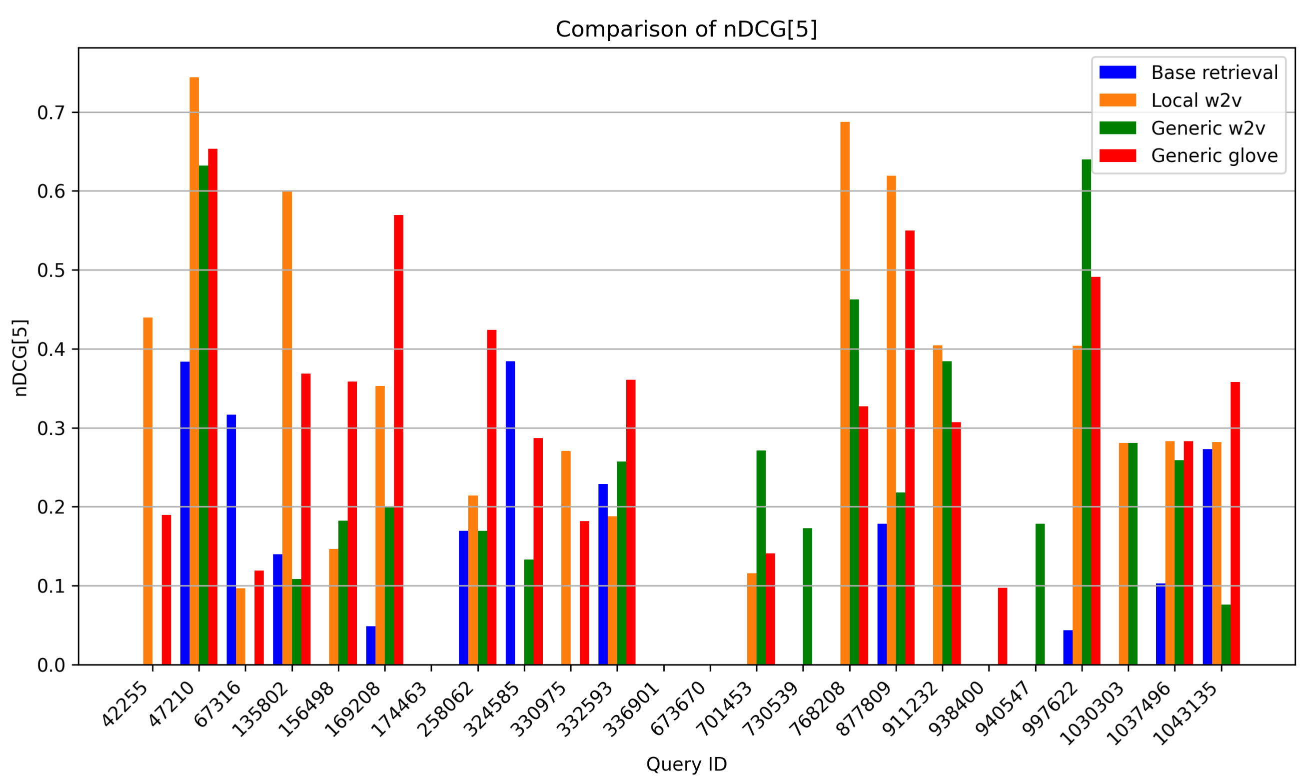

Figure 2: Comparison of `nDCG[5]`

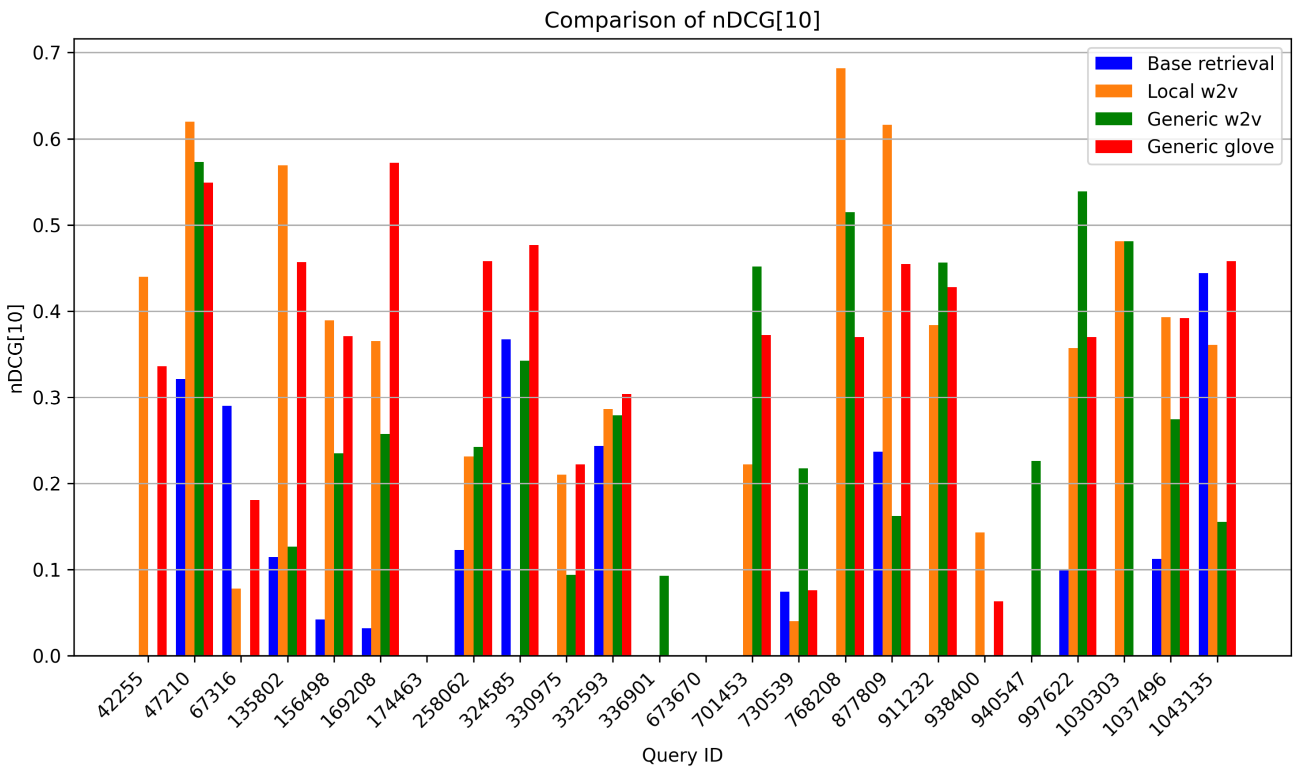

Figure 3: Comparison of `nDCG[10]`

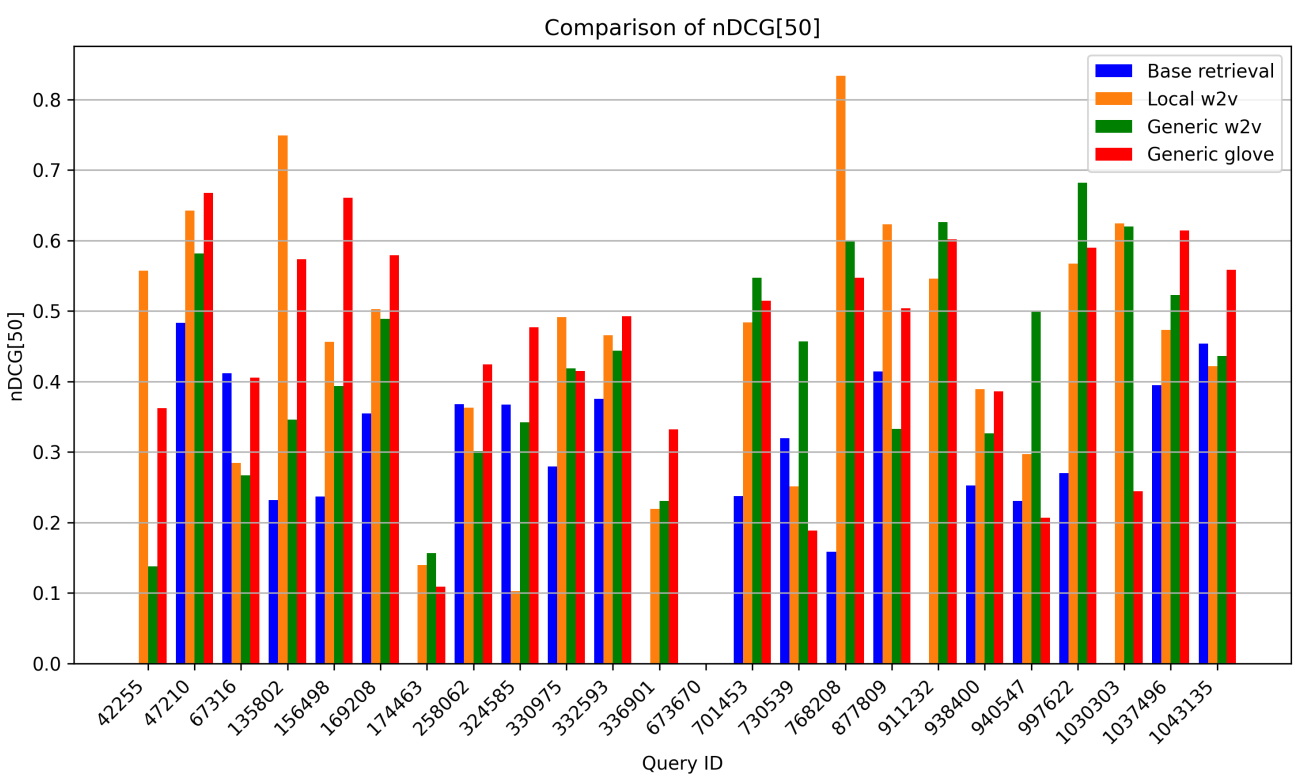

Figure 4: Comparison of `nDCG[50]`

List of Tables

1 Model Hyperparameters 3
2 Intermediate vocabulary sizes of each model 4
3 nDCG[5] values for Local w2v 4
4 nDCG[10] values for Local w2v 5
5 nDCG[50] values for Local w2v 5
6 nDCG[5] values for generic w2v 6
7 nDCG[10] values for generic w2v 6
8 nDCG[50] values for generic w2v 7

List of Figures

1 Training loss for queries after each epoch 3
2 Comparison of nDCG[5] 7
3 Comparison of nDCG[10] 8
4 Comparison of nDCG[50] 8