Methods and Initial Results

Reid McIlroy-Young

May 17, 2017

For my project I will be using the Web of Science (WOS) database hosted by Knowledge Lab. It has metadata on almost all scientific publications from 1960 to 2015, with new records being more complete. Each publication can be linked to one or more other tables each which contain other metadata than the main table, the number of entries for each table I am concerned with are shown in figure 1 and the complete database schema in figure 2. Access to the database is controlled by Knowledge Lab so they would need to be contacted to access it, once access rights are obtain the database is found at wos2.cvirc91pe37a.us-east-1.rds.amazonaws.com and the documentation at http://docs.cloudkotta.org/dataguide/wos.html.

The data for WOS were collected by Thompson Reuters until 2016, when it was given to Clarivate Analytics who now maintain it. The contemporary publications are collected from the publishers directly while older and more obscure publications are obtained from scanned copies digitalized with OCR, which is one of the factors that leads to newer publications having much higher quality data.

Table	Number of Entries
publications	57136685
abstracts	26093439
publishers	50668193
keywords	78155603
references	1085738245
years	75

Figure 1: *Web of Science database statistics*

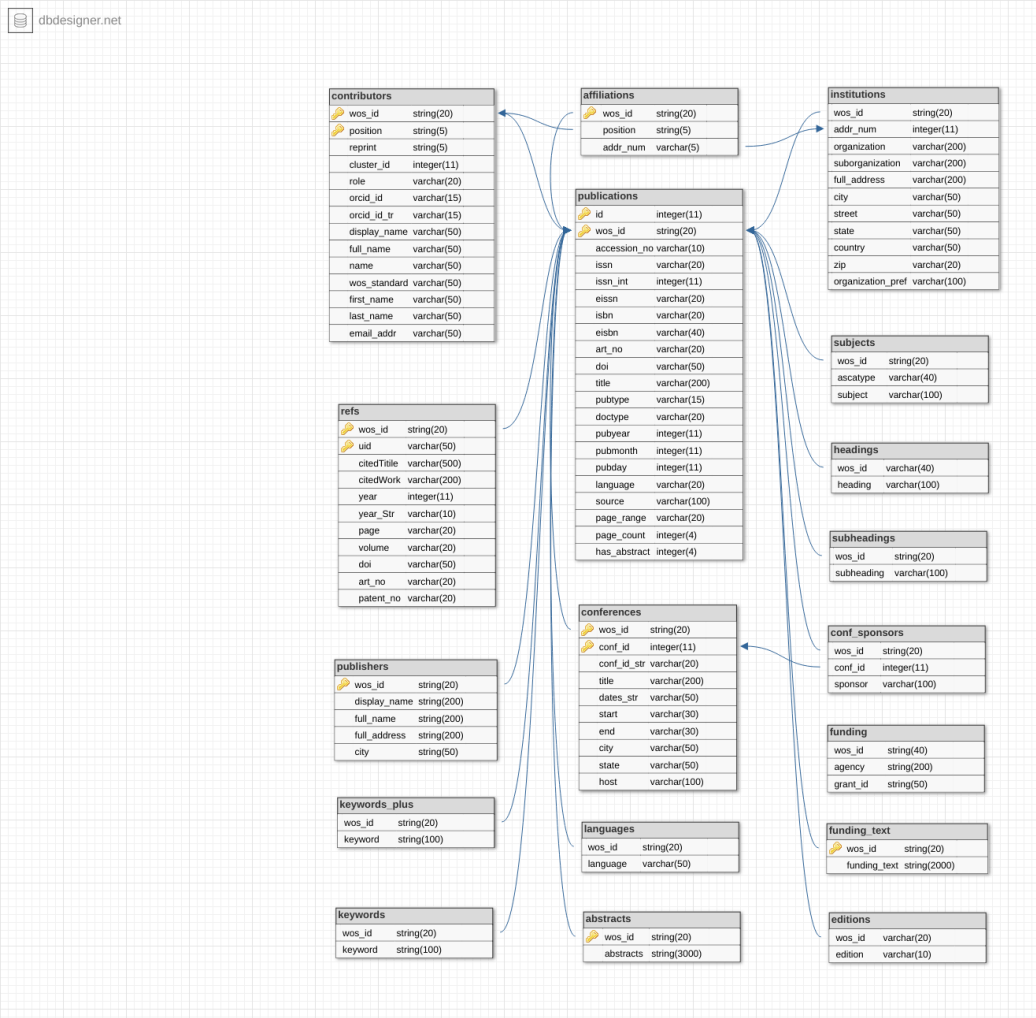

Figure 2: *Knowledge Lab WOS database schema*

Once I have the data collected I divide it into classified (training) and unclassified publications. Some journals almost entirely publish new scientific software so all articles from these can be classified as containing new software. There are also some that contain virtually none, all of their publications can also be classified as negative. Then there are journals that can contain both, these I will be hand coding samples from to provide my model a more varied set of inputs. Since the end goal is to not rely on anything but the title and abstract I have only coded a small number of journals and will only do enough to get my model working. The classified journals are given in figure 3, note they are all top contemporary statistics journals, since that is what I am limiting my initial analysis to. Once a training set has been generated

I can give the abstracts, titles and classifications to my model for training.

Journal	Classification	Total Citations	Impact Factor
R JOURNAL	Mostly Software	271	1.045
STATA JOURNAL	Mostly Software	2636	1.292
JOURNAL OF STATISTICAL SOFTWARE	Mostly Software	6868	2.379
STATISTICS AND COMPUTING	Some Software	2682	1.786
ADVANCES IN DATA ANALYSIS AND CLASSIFICATION	Some Software	214	1.707
ECONOMETRICA	Little Software	24957	4.053
TECHNOMETRICS	Little Software	6062	1.435
STATISTICAL METHODS IN MEDICAL RESEARCH	Little Software	2703	4.634

Figure 3: *Classified journals, citations and impact factors are both from 2016 (yes the all caps is ugly, but that's how WOS titles them)*

The first step is tokenizing the abstract and title, this is done with *nltk*'s word and sentence tokenizers. I am still determining if stemming is helpful so I may add it to the pipeline as well, again with an *nltk* function. The next step is converting to word embedding vectors, this is done with word2vec, using Google's pre-trained model, with missing words being marked with a special vector and not being discarded, since they are often going to be the titles of the software packages. I have taken care to make both these steps deterministic so that their variances are not factors in the neural network.

Once the texts have been converted into sequences of vectors they can be given to the neural network, for this analysis I am using a recursive neural network composed of two Bidirectional Recurrent Neural Networks (BRNN), one for each of the inputs, followed by a fully connected $ReLu$ layer (u). The BRNNs are composed of four layers (each 64 to 256 nodes): forward (h), backward (g), output (o) and collection (c). The first three layers have $tanh$ activations while the last uses $softmax$. The network is trained with backpropagation using stochastic gradient descent. Since the abstracts can be quite large (30 000 characters) vanishing/exploding gradients are an issue so Long Short Term Memory (LSTM) will be used to augment the abstracts BRNN. The design of the model is laid out in figure 5. I do not yet have a working classifier so cannot give it's statistics, but that is mostly due to the temporal limitation of training time and as such I am confident I will have one working with in the next couple weeks.

Once the publications introducing new scientific software have been identified I will be looking at their embeddings in their fields. I have generated few networks from my training data (all 'Mostly Software'), all publications from the journal up to 2015, as an example, but the final analysis will have a larger scale. Figure 4 shows the summary statistics for the networks. As you can see from the table, despite the journals have varying network sizes, both their transitivity and densities are similar when the same network types are constructed, thus there is hopefully some homology between them meaning the final results will be detecting members of a larger pattern and not random distributed nodes.

Journal	Network	density	edges	isolates	loops	nodes	transitivity
Combined	Cocitation	0.001432	1168336	9	246	40402	0.314339
	Coauthorship	0.000765	4384	256	0	3385	0.565859
	Citation	0.000038	65864	0	22	41542	0.004958
	Keyword	0.004871	22325	47	0	3028	0.134333
R JOURNAL	Cocitation	0.006672	67446	4	22	4497	0.827388
	Coauthorship	0.004224	570	65	0	520	0.865347
	Citation	0.000222	4930	0	0	4714	0.000275
	Keyword	0.017685	858	11	0	312	0.582090
ANNALS OF STATISTICS	Cocitation	0.003099	613206	4	143	19893	0.311618
	Coauthorship	0.001816	1897	63	0	1446	0.369824
	Citation	0.000086	35826	0	18	20366	0.006957
	Keyword	0.008447	15561	13	0	1920	0.160793
JOURNAL OF STATISTICAL SOFTWARE	Cocitation	0.002738	492165	1	87	18961	0.400034
	Coauthorship	0.001607	1952	155	0	1559	0.790647
	Citation	0.000066	25108	0	4	19467	0.002048
	Keyword	0.006545	6465	42	0	1406	0.214815

Figure 4: *Summary statics for journal networks, 'Keyword' means a keyword concurrence network*

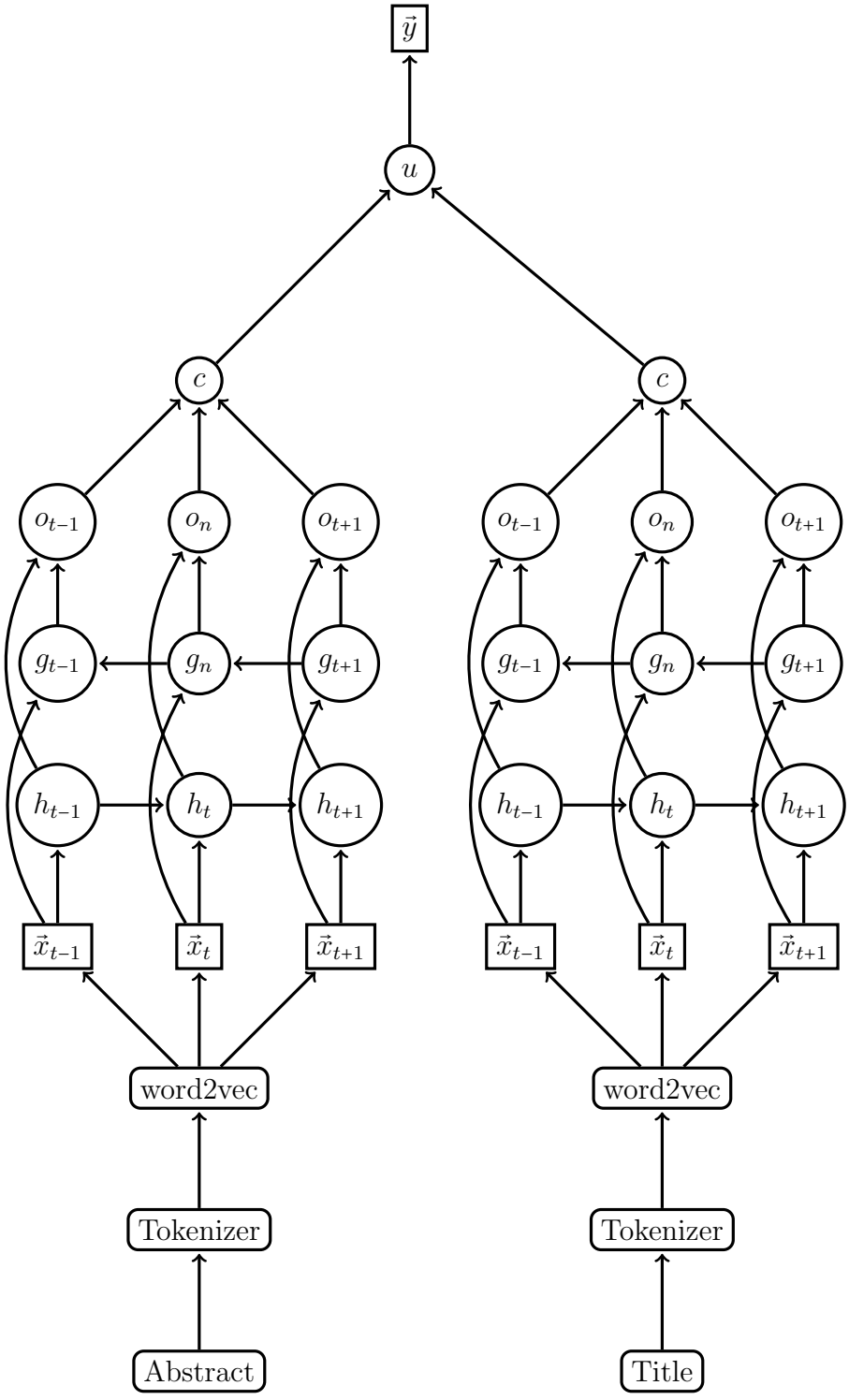

Figure 5: *Simplified recursive BRNN layout, note the lack of LSTM. Circles are NN layers, rectangles are the preprocessing, and the finale output and NN inputs are vectors, indicated with square corners*