UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

FIB Facultat d'Informàtica de Barcelona

Word Embeddings and Topic Detection for Contextual LSTM NLP tasks: Part II

Authors:
Pol Alvarez Vecino

May 24th 2017, Barcelona

Contents

1 Introduction **2**

2 Requirements **2**

3 Topic extraction **2**

3.1 Latent Dirichlet Allocation 3

3.2 Latent Semantic Indexing 4

3.3 Hierarchical Dirichlet Process 4

4 Context construction **4**

5 Words to IDs validation **5**

6 Dataset **5**

7 Results **5**

8 Conclusions **8**

1 Introduction

This the second part of the project aiming to prepare a dataset to train a Contextual Long-Short Term Memory (CLSTM) neural network as described by Ghosh et al. [1]. In order to be used with CLSTM, the dataset needs the words' embeddings and be able to produce a topic/context for a subset words. To give more insight to the decisions made throughout this deliverable, the work required to actually train the CLSTM network has been divided into the following objectives:

1. Preprocess dataset for word embeddings creation
2. Train the word embeddings
3. Use word embeddings IDs to transform wikipedia articles into lists of embeddings' indices
4. Train a LSTM network for perform word prediction (and act as a result baseline)
5. **Find a way to provide the some kind of context the CLSTM**
6. **Develop an efficient mechanism for querying the context during network training**
7. Train CLSTM network for word prediction

In this second deliverable we will focus on the topic construction and querying. First, we will lay out the type of inputs our model will need to give context to. After, we will explore different methods of constructing the "context" for a document, paragraph, and sentence and how well they match our needs.

2 Requirements

To use them as context we need to get a topic of an arbitrary-long word sequence. This added context must be fast to produce to avoid slowing down the already slow training process of a deep neural net. It must also do not require too much computation or leave a big memory footprint as it also can hinder the training process.

If the method is either too slow or too costly we will try to annotate directly the dataset with the topics. Unfortunately, this will add another layer of complexity (we will need to split again the dataset to obtain the labels - i.e. the next word to predict - without the topic), will cause the dataset size to grow considerably, and will force us to pre-process the whole dataset each time we want to make some change to the topics used.

Taking into account these considerations, out context technique should be able to:

- Evaluate context of a sequence of words with arbitrary length
- Reasonable memory footprint
- Fast context computation
- Produce sensible contexts

3 Topic extraction

Associated files:

· **topics_analysis.py,** handles the creation/loading of the required inputs for each of the topic models.

In order to provide context for the posterior CLSTM implementation the first approach was to use topics. These methods are trained upon a corpus of documents and can then produce a topic for new documents (of arbitrary length). They fulfill all our requirements out-of-the-box quite well (except of sensible topics which

will need to manually checked). In exchange for this capabilities though, they have a long training time so we will need to see if they can produce sensible topic models in a reasonable time.

I explored the most frequently used algorithms for topic detection and evaluated their performance on large datasets. Specifically, the topic detection algorithms used were:

- Latent Dirichlet Allocation online [2]
- Latent Dirichlet Allocation offline
- Latent Semantic Indexing
- Hierarchical Dirichlet Process

All methods required a corpus (bag of words of each document with words IDs) and the corresponding dictionary with the word-¿ID mapppings. Some preprocessing is done when constructing the dictionary such as filtering the 10% most frequent words or keeping the total dict. size ¡ 200.000. On the first approach I manually built that corpus and mappings prior to the analysis. This worked well for local development with a reduced subset of files. When using the whole Wikipedia, this "naive" approach was not able to finish in under 35 hours (which was my maximum allowed job time). Instead, I used an already optimized gensim function

```
python -m gensim.scripts.make_wiki /path/to/wikipedia/data.gz
```

which converts the Wikipedia dump (without even needing to uncompress it) and creates the word-ID mappings and the corpus of the data as compressed TF-IDF vectors that can be fed to our topic models.

3.1 Latent Dirichlet Allocation

Latent dirichlet allocation (LDA) is a generative probabilistic method that represents sets of samples to be described by possibly unobserved groups that describe why some parts are similar. When it applied to topic detection it models each document with a finite mixture of a set of topics. The topics, in turn, are an inifinite mixture of a set of probabilities. The model was proposed independently by Pritchard et al. [3], applying it to population genetics and by Blei et al. [4] for topic discovery.

The model is trained upon corpus of documents and then be used to predict the topics of new sequences. It is a good fit for our requirements but it requires that the number of topics must be specified beforehand. Unfortunately, this adds another hyperparameter to be tuned in order to find the best topics (potentially the whole topic model should be used for CLSTM training in order to choose the best).

The library *gensim* uses a fast implementation of online LDA parameter estimation based on [2]. We did many trials in order to check which of the available running settings were best. Basically, we first run the simple or batch LDA (not online) which iterates over all the corpus each time before updating the weights. This method was too slow so we tried online LDA, in which weights are update after a given number of documents. If documents do not show much topic drift (as it would be the expected situation for Wikipedia) the smaller update are quite good and the model tends to converge faster. Finally, we also used the multicore version of batch LDA which we managed to train for 3 whole passes using 12 cores.

Training times for the multicore version were around 14.5 hours. The single-pass online LDA execution took 12 hours. Both were run with 100 topics (following "advice" from *Experiments on the English Wikipedia* [1].

[1] https://radimrehurek.com/gensim/wiki.html

3.2 Latent Semantic Indexing

Latent semantic analysis or indexing is similar to LDA, it tries to relate documents and their elements with a set of concepts. LSI uses singular value decomposition (SVD) to reduce the matrix containing the documents' word counts (rows are different words and columns represent the documents' index) by removing rows while preserving column similarity. With this matrix, distance between is computed with cosine of their vectors' angles. Finally, LSA groups together documents with similar words as well as words appearing in similar document to detect the latent concepts which will be used as topics.

LSA results are more difficult to interepret than LDA because each one of them is a weighted sum of words (both positive and negative) instead of probabilities. The average training time for LSA was about 4 hours, quite lower than its LDA counterpart.

3.3 Hierarchical Dirichlet Process

The hierarchical Dirichlet process (HDP) is a nonparametric (does not need the number of topics!) bayesian method to cluster data. It uses a Dirichlet process for each document and all of them share the same base distribution which is in turn drawn from a Dirichlet process. Published by Teh et al. [5], it is a generalization of the inifinte hidden Markov model published in 2002 by Beal et al. [6]. Gensim uses an online variation [7] of the original algorithm developed for very large datasets (which become too slow because posterior inference algorithms require multiple passes of all data).

HDP was the best promise because it does not require to specify the number of attributes and the results are easy to interpret (each word has a probability associated). Unfortunately, gensim version is not throughly tested and it was impossible to get meaningful results out of all the Wikipedia data.

4 Context construction

· **creator.py,** implements the context creation using LDA best word, LDA average and LSI sum methods (described below)

The ultimate goal of the topic extraction is to pass it along with each word a context. We are interested in two kinds of contexts:

SentSegTopic:
sentence segment topic, which is uses as context the topic of the current sentence segment read so far.

ParaSegTopic:
paragraph segment topic, which is uses as context the topic of the current paragraph segment read so far.

For both types what we actually need as output is a a vector with word-embedding. To do so, many approaches can be followed.

For LDA we used two different methods to create the context word embedding. First was to simply use the word embedding corresponding to the most important word of the topic. We decided to try this simple approach because the small trials done with the LDA model reported sensible words for sentence and paragraphs segments (see Listing 1). Second method used was to do weighted sum (using their probabilities as weights) of the first ten relevant words for the topic of the segment. This second method was added to check if some relevant information that one-word per topic technique could not provide is important for the word prediction task.

For LSA we decided to use the sum of the all words belonging to the topic because the number of words per topic is fixed. The weights are not probabilities so we are not just finding the "center" of the topics averaged by their relevance, instead, this is more similar to performing semantic arithmetics on the embedding space as we did in previous work (where for example King+Woman-Man=Queen). Because of that, the interpretability of each of the topics is quite more difficult to assess than the LDA results.

Finally, we will also consider another method which does not require any further computation: averaging the embeddings read so far. For this technique we will start with the embedding of the first word as context. Next, for each new word we will average the current context with the next word embedding. We will try two approaches: one in which every time we compute the whole average so that all words have the same weight; on the other, we will just do the mean of the context and the new word, this will have the effect of overweighting recent words and making the older ones vanish. Both methods will be reset for each new sentence (when computing the SentSegTopic) or each paragraph (when computing the ParaSegTopic). Probably, averaging all the words will be better because we are working with sentences and paragraphs (which should be short) and having the focus always on the most recent words does not seem useful.

5 Words to IDs validation

Associated files:

· **embeddings.py,** handles the preprocessing of the "wikiExtracted" data and creates the embeddings with it

As stated in Future Work of previously deliverable, before starting to use all these time-consuming methods upon transformed data it was almost mandatory to ensure the correct word to ID translation. For that, I developed an script which does the reverse translation and checks that it matches with the original document. This will not yield exact matches because of the unknown words so the threshold for similarity can be tuned and the script outputs both the percentage of similarity as well as the ratio of unknowns.

6 Dataset

The dataset used is the Wikipedia English corpora as of 20th of February 2017. It is a full raw dump so it contains HTML tags, links and many other useless informations that need to be stripped.

- Compressed data size: 13 GB
- Uncompressed data size: 57 GB
- wikiExtracted data size: 12 GB
- Number of documents: 5.339.722

7 Results

Associated files:

· **test_topics.py,** runs the tests presented in this section with the LDA and LSI models

Analyzing the results of semantic and unsupervised methods is not easy. Specially because there is best representation *per se*, it depends on what we try to achieve. In this project, best evaluation would be the results of the training of CLSTM. However, we can inspect a bit the results to see if they are producing expected topics for some short fragments. To do this trials we have used the next five demo fragments:

Demo 1:

the roman consul is normally a notable person from the senate elected by direct voting of the italic tribes

Demo 2:

agriculture is the cultivation and breeding of animals plants and fungi for food fiber biofuel medicinal plants and other products used to sustain and enhance human life

Demo 3:

anarchism is a political philosophy that advocates societies based on voluntary institutions

Demo 4:

the economy of honduras is based mostly on agriculture which accounts for of its gross domestic product gdp in 2013

Demo 5:

hermann wilhelm or goering 12 january 15 october 1946 was a german politician military leader and leading member of the nazi party nsdap

Listing 1 and 2 show the results of LDA in batch and online mode respectively. We see that results are quite similar except in Demo 4 in which online version outputs some strange topics related to Poland. Possibly, there is some topic drift in the way we fed the data to the online LDA. Taking into account, that online version is normally used when faster results are needed, we decided to use the parallel version for the future CLSTM training. We also see that using first word's topic embedding as context seems quite reasonable because the word is a good topic for each sentence.

Listing 1: Topic test with LDA multicore implementation with 100 topics and 3 full passes. Ten words per topic are shown to check if doing the average of the first ten words makes sense.

```
[BLOCK] Loading dictionary files from ../models/topics/gensim_wordids.txt.bz2

Demo 1:
the roman consul is normally a notable person from the senate elected by direct
voting of the italic tribes
[u'party', u'election', u'minister', u'elections', u'council', u'elected',
u'parliament', u'votes', u'assembly', u'liberal']

Demo 2:
agriculture is the cultivation and breeding of animals plants and fungi
for food fiber biofuel medicinal plants and other products used to sustain and
enhance human life
[u'food', u'restaurant', u'wine', u'chef', u'beer', u'cuisine', u'dish',
u'milk', u'brewery', u'meat']

Demo 3:
anarchism is a political philosophy that advocates societies based on voluntary
institutions
[u'social', u'philosophy', u'theory', u'research', u'psychology', u'human',
u'book', u'how', u'studies', u'study']

Demo 4:
the economy of honduras is based mostly on agriculture which accounts for of
its gross domestic product gdp in 2013
[u'village', u'municipality', u'district', u'population', u'town',
```

```
u'settlement', u'oblast', u'region', u'administrative', u'census']

Demo 5:
hermann wilhelm or goering 12 january 15 october 1946 was a german politician
military leader and leading member of the nazi party nsdap
[u'german', u'der', u'von', u'und', u'germany', u'berlin', u'die', u'hans',
u'austrian', u'vienna']
Total processing time: 2 seconds

Total processing time: 1 seconds
```

Listing 2: Topic test with LDA online implementation with 100 topics and 1 full passes. Again

```
[BLOCK] Loading dictionary files from ../models/topics/gensim_wordids.txt.bz2

Demo 1:
the roman consul is normally a notable person from the senate elected by direct
voting of the italic tribes
[u'party', u'election', u'minister', u'assembly', u'elected', u'council',
u'elections', u'legislative', u'parliament', u'politician']

Demo 2:
agriculture is the cultivation and breeding of animals plants and fungi for
food fiber biofuel medicinal plants and other products used to sustain and enhance
human life
[u'protein', u'cell', u'gene', u'cells', u'acid', u'virus', u'drug',
u'proteins', u'bacteria', u'dna']

Demo 3:
anarchism is a political philosophy that advocates societies based on
voluntary institutions
[u'book', u'social', u'language', u'text', u'philosophy', u'how', u'meaning',
u'theory', u'human', u'culture']

Demo 4:
the economy of honduras is based mostly on agriculture which accounts for
of its gross domestic product gdp in 2013
[u'polish', u'poland', u'warsaw', u'berghahn', u'krakow', u'bock', u'bydgoszcz',
u'sv', u'poznao', u'stanislaw']

Demo 5:
hermann wilhelm or goering 12 january 15 october 1946 was a german politician
military leader and leading member of the nazi party nsdap
[u'german', u'der', u'von', u'und', u'berlin', u'germany', u'die', u'hans',
u'austrian', u'karl']
Total processing time: 2 seconds
```

Listing 3: Topic test with LSI and 400 topics. It is difficult to appreciate if the topics are actually sensible because of the weighted sum of the words. We see some clues like "roman" and "king" words appearing in first

sentence but also "ohio" and "hotel" are there.

```
[BLOCK] Loading dictionary files from ../models/topics/gensim_wordids.txt.bz2

Demo 1:
the roman consul is normally a notable person from the senate elected by direct
voting of the italic tribes
((197, 0.16942229990484226), u'0.323*"smith" + -0.189*"pennsylvania" + -0.161*"ohio"
+ -0.158*"hotel" + -0.153*"king" + 0.139*"illinois" + -0.136*"puerto" +
0.135*"carolina" + -0.127*"san" + 0.124*"roman"')

Demo 2:
agriculture is the cultivation and breeding of animals plants and fungi for
food fiber biofuel medicinal plants and other products used to sustain and
enhance human life
((83, 0.21615013437926739), u'-0.189*"railway" + 0.160*"plant" +
-0.139*"minister" + -0.136*"park" + -0.136*"commune" + -0.130*"km" +
0.129*"station" + -0.123*"hospital" + 0.122*"japan" + 0.115*"japanese"')

Demo 3:
anarchism is a political philosophy that advocates societies based on
voluntary institutions
((292, 0.093584051214839425), u'0.205*"beach" + 0.102*"science" +
0.101*"jersey" + -0.098*"minnesota" + -0.096*"oil" + 0.094*"ontario" +
0.094*"kansas" + -0.090*"bangladesh" + -0.087*"field" + 0.085*"data"')

Demo 4:
the economy of honduras is based mostly on agriculture which accounts
for of its gross domestic product gdp in 2013
((379, 0.077513235844465997), u'-0.169*"seine" + 0.138*"climate" +
-0.124*"maritime" + -0.115*"fish" + 0.099*"software" + 0.096*"soviet" +
-0.094*"rock" + 0.092*"calais" + 0.090*"term" + -0.089*"network"')

Demo 5:
hermann wilhelm or goering 12 january 15 october 1946 was a german
politician military leader and leading member of the nazi party nsdap
((5, -0.3574736501402212), u'-0.286*"party" + -0.285*"district" +
-0.255*"election" + -0.240*"village" + -0.137*"municipality" +
-0.135*"church" + -0.135*"album" + -0.114*"elections" + -0.108*"elected"
+ -0.103*"league"')

Total processing time: 1 seconds
```

8 Conclusions

We have explored four different methods to provide context for the LSTM. First method returns the embedding of the best LDA topic word. This is the favourite because it is easy to use and it only requires a dictionary lookup operation (for the embedding) a part from the actual model inference.

If this first method is losing too much information by throwing away the rest of words of the topic we will try to use the second method were we average the first 10 topic words. However, this will probably add an extra overhead to the model or add words not too relevant to the context.

Third method, using the sum of the LSA words for the best topic, is quite difficult to analyse in advance as it returns an embedding vector. Again, this method should add some extra overhead but it should not be too big so it will also be tested.

Finally, last method adds the embeddings read so far in order to built kind of a semantic context by averaging the meaning of the words seen so far with the next one.

References

[1] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, "Contextual LSTM (CLSTM) models for Large scale NLP tasks,"

[2] M. D. Hoffman, D. M. Blei, and F. Bach, "Online Learning for Latent Dirichlet Allocation," *Advances in Neural Information Processing Systems*, vol. 23, pp. 1–9, 2010.

[3] J. K. Pritchard, M. Stephens, and P. Donnelly, "Inference of Population Structure Using Multilocus Genotype Data," *Genetics*, vol. 155, no. 2, 2000.

[4] D. M. Blei, B. B. Edu, A. Y. Ng, A. S. Edu, M. I. Jordan, and J. B. Edu, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[5] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet Processes," *Computer*, vol. 101, no. 476, pp. 1–41, 2005.

[6] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, "The infinite hidden Markov model," *Advances in Nueral Information Processing Systems*, vol. 14, pp. 577–584, 2001.

[7] C. Wang, J. Paisley, and D. M. Blei, "Online Variational Inference for the Hierarchical Dirichlet Process," *Icais*, vol. 15, pp. 752–760, 2011.