Building a topical representation of words

Done by Andrei Constantinecu At the University of Pennsylvania – August to December 2017

Under the supervision of Professor Ani Nenkova

0. Introduction and motivation

How to best represent words has been a very prolific topic in Natural Language Processing. The task is come up with a way of representing words that capture relationships between them, such as synonymy, antonymy, relatedness, etc.

In the recent year, distributed representations of words, or Word2Vec introduced by Mikolov et al. have gained a lot of popularity. The idea being to represent words as vectors in a small dimensional space, where the vectors representing words are obtained by running a neural model.

Here we propose another representation of words, a topical representation of words. We will also represent word as vectors, but this time dimensions will have a clear meaning and interpretation. It quantifies the strength of association of a word to a topic. We hope that this more interpretable representation of words comes with good and practical uses and properties.

1. Topical representation of words

We seek to represent each word as a vector whose dimensions reflect the strength of association of a word to a series of topics. Let us say we have 12 topics: Arts - Sports – US – Style – Health- Travel - Business – World - Movies – Technology – Education – Science. The first dimension in this representation of the word will reflect how associated the word is to the topic Arts, the second to the topic Sports, and so on.

From this idea, one might expect to obtain the following kind of representations:
Painter = [10 0 0 0 0 0 0 0 0 0 0 0]
Baseball = [0 10 0 0 0 0 0 0 0 0 0 0]
Where the word ‘painter’ would be heavily associated to the topic ‘Arts’, and the word ‘Baseball’ to the topic ‘Sports’.

1.1 The Data and method

To compute topical scores for each word and for each topic, we will use a New-York Times annotated corpus, and we will contrast distributions of words between a topical corpus and a reference corpus. This corpus comprises all New-York times articles published between 1987 and 2007. The articles are annotated with several pieces of metadata such as the section in which they appeared, or descriptors specifying in more detail what an article is about. For each topic, we therefore have a topical sample of articles, i.e. articles being published in the same section of the newspaper or articles being described with the same descriptor, and a reference corpus. As a reference corpus, we ran through the entire New-York-Times corpus and computed frequencies for each word.

1.2 Five techniques to compare differences in distributions

1- Log odds: We compute a score for each word as follows:
Score_log_odds(word) = log [(frequency topic) / (frequency reference)]
This technique is equivalent to computing the Point-wise mutual information between a word and a topic.

2-Signature: This technique is based on a statistical test.
If we set H1 as the hypothesis that the distribution of the word in the topic corpus is not different from that of the reference corpus, and H2 as the hypothesis that the two are different, Lin et al. showed that:

$$-2log\frac{L(H_1)}{L(H_2)}$$

follows a chi square distribution. The higher the chi-square statistic the more likely the two distributions are different, the more likely the word is a topical word.

3- Regression delta: We compute a score by devising a regression task. If we consider a regression task where the aim is to predict the frequency of a word in the whole corpus given its frequency in a biased topical corpus, one might imagine that the prediction of topical words will overshoot. Thus, we can compute a score for each word as follows:
Score_regression(word) = Prediction(word) – Reference(word)

4 - Bag of words: The ideas is to train a classifier on bag of words features to predict the topic of a specific article. Then, one can use the weights outputted by the classifier for each word as the topical score of words. For instance, if the word 'painter' appears only in articles labeled as belonging to the topic 'Arts', one might expect the classifier to assign a high score to the word painter, since it is a useful discriminating feature. As such it would be a good candidate as a topical word.

5- LDA: The LDA model (Latent Dirichlet Allocation) is completely different from all the ones considered so far. It is an unsupervised algorithm that infers its own topics, and assigns scores for each word in each topic. Therefore, it would be interesting to compare how our hand-labeled topics compare to LDA ones.

1.3 External task: Topic classification on the Reuters dataset

To evaluate each technique, we shall use an extrinsic task. The Reuters dataset is arguably the most used dataset for topic classification. It is comprised of 21000 articles, already divided into training and testing sets, with 90 different topics, each article belonging to at least one topic, possibly more.

We will use the techniques mentioned above, i.e. log odds / signature / regression, bag of word weights, and LDA weight for 100, 200, and 300 topics or dimensions. These topics correspond to descriptors of the New-York Times, which can be described as content tags. A list of these descriptors can be found in the data folder. These descriptors were ordered by frequency of appearance in the New-York Times. We ensured that these descriptors were not redundant by merging descriptors which shared more than 50 % of their articles. These descriptors will become our topics.

The features we feed our classifier (Linear SVC) are the mean representation of a text, i.e. we average the representation of each word in the article. As a first baseline we included a bag of words model. Also, we included more widely-used distributed representation of words embeddings.

Model	Micro F1	MacroF1	Recall micro	Precision micro
BOW	0,86	0,48	0,82	0,91
Signature 100	0,39	0,154	0,572	0,302
LDA 100	0,41	0,11	0,54	0,33
Regression 100	0,497	0,177	0,613	0,421
BOW from NYT 100	0,677	0,239	0,586	0,803
log odds descriptors 100	0,798	0,319	0,734	0,874
Glove 100	0,799	0,341	0,735	0,875
Signature 200	0,49	0,201	0,705	0,385
LDA 200	0,462	0,18	0,602	0,375
Regression 200	0,608	0,22	0,696	0,545
BOW from NYT 200	0,727	0,302	0,642	0,8375
log odds descriptors 200	0,82	0,397	0,778	0,867
Glove 200	0,826	0,399	0,779	0,879
Signature 300	0,532	0,235	0,736	0,417
LDA 300	0,513	0,27	0,657	0,421
Regression 300	0,66	0,241	0,572	0,781
BOW from NYT 300	0,775	0,343	0,698	0,872
log odds descriptors 300	0,824	0,421	0,792	0,858
Glove 300	0,837	0,429	0,793	0,886
Word2Vec 300	0,84	0,397	0,78	0,901

One can see that the log odds techniques performs the best, and gives comparative results to widely used word embeddings such as Glove or Word2Vec. We conclude that the log odds technique is the best way to assign topic scores to words.

1.4 Extending the lexicon

Until now we have not addressed the issue of coverage. Our topical representation of words represents each word as a d-dimensional vector, where each dimension represents a score reflecting the association of a word to a topic. Nevertheless, our lexicon suffers from two coverage problems. It does not include enough words, and for the words it includes, we do not have scores for all dimensions. Although the New-York Times is big, one notices that it is by no means exhaustive. The Glove representation offers representations for 400 000 words, and Word2Vec for 3 million words. Our lexicon offers only representations for 50 000 words. What is more, these representations are often incomplete. If in a specific topic, i.e. for a specific dimension, a word has not appeared a sufficient number of times, or has not appeared at all, there is no topical score for this word in this particular topic. On average each word has only 40 % of its dimensions covered. When we did not have a score for a dimension of a word, we conveniently addressed this problem by setting its value to 0. Nevertheless, one can see the

obvious utility of obtaining more refined score for each dimension. This would allow our lexicon both to extend to more words and to be more complete.

To achieve this extension, we took as features glove embeddings, as target the log odds score of our seed words, and trained a regularized regression model (Ridge). The average r-square obtained for our topics is .84. This result is encouraging, meaning that it is possible to make a more complete version of our topical lexicon using this framework. Glove embedding has been found to work better than Word2Vec embeddings for this task, although by a little margin.

To prove the usefulness of this framework, we applied our augmented topical representation of words to the text classification task, and obtained encouraging results that can be found bellow.

Model	Micro F1	MacroF1	Recall micro	Precision micro
BOW	0,86	0,48	0,82	0,91
log odds descriptors 100	0,798	0,319	0,734	0,874
log odds descriptors 100 augmented	0,797	0,3714	0,746	0,856
Glove 100	0,799	0,341	0,735	0,875
log odds descriptors 200	0,82	0,397	0,778	0,867
log odds descriptors 200 augmented	0,825	0,399	0,776	0,881
Glove 200	0,826	0,399	0,779	0,879
log odds descriptors 300	0,824	0,421	0,792	0,858
log odds descriptors 300 augmented	0,839	0,442	0,805	0,876
Glove 300	0,837	0,429	0,793	0,886
Word2Vec 300	0,84	0,397	0,78	0,901

1.5 Example of topical words that are out of vocabulary

Once we trained a model for each topic, it is possible to use this model to predict which are the most topical words. Taking all the words indexed in Glove one can obtain scores for words that are out of vocabulary, but for which we have a word embedding. Thus, one can sample for each topic the top-k words as outputted by the model. Below are some results:

We notice encouraging results. This sample of words with highest scores is consistent with what we would expect. We obtain words that are heavily linked with a specific topic, such as genre movements for Arts, or company names for Business. This result echoes the general behavior of the log odds scoring method. Only words that are specifically linked with a topic appear in this list, instead of common words that are more used in a specific context.

1.7 Comparison to Glove

A relevant question to answer now is to what extent this topical representation of words captures information that is not included in Glove. As the performance of the two in the topic classification task is so similar, is there any added value in the topical representation, or is it redundant with the glove one. What kind of behavior does one representation capture as compared to another?

To that aim, we will use two comparisons. The first is looking at the nearest words, the second is seeing how a concatenation of the two combinations work in the Reuters classification task.

1.7.1 Closest words

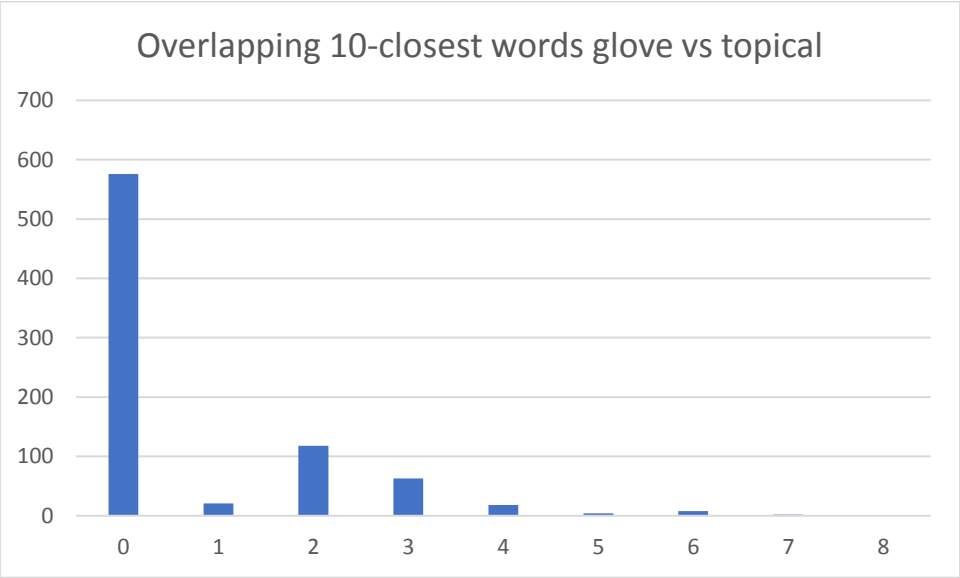
First, one can get a qualitative sense of the difference between the two techniques by looking at the 10 closest words to a list of random words.

Word	Closest topical NYT	Closest Glove
detained	detention,jailed,policemen,detain,interrogated,,deported,tortured,clashes,grenade,imprisoned	Detention,,jailed,arrested,detain,arrest,custody,suspects,suspected,deported,police
hatcher	griffins,benched,krzyzewski,rupp,bibby,alamodome,smoltz,sanderson,popovich,dunleavy	defenseman,roenick,puck,defensemen,jere,niedermayer,leetch,goalie,shanahan,bourque

postgame	vinny,strahan,illini,marv,favre,telecasts,stats,schaap,halas,spurrier	pregame,locker,postfight,impromptu,halftime,predraft,celebratory,preseason,prefight,handshakes
coli	salmonella,o157,h7,escherichia pathogens,contamination,bacterium, ,contaminants,arsenic	Salmonella,o157,bacteria,,h7,escherichia,0157, bacterium,listeria,bacterial,staphylococcus
baroque	gothic,impressionist,majestic,exquisite,orchestral, ballroom,elegance,eclectic,stylish,decorative	Gothic,rococo,romanesque,renaissance,neoclassical,medieval,classical,painter,italianate,symphonic
silvera	dokes,molotsky,amadiya,kisito,varrone,vejar,alcivar,falin,ybarra,follet	cartolini,bustani,minnig,mosquera,bajarin,sharmba,yasushi,hubner,cerkez,sifford
serves	Serve,meals,,menu,tables,soup,accessible,dishes,lunch,meal,chairs	serve,serving,served,operates,located,provides, unincorporated,district,county,grades
coordinate	coordinating,coordination,undermined,alarmed,undermining,convened,undertaken,concerted,complicating,combating	coordinating,coordination,coordinates,formulate,coordinated,oversee,cooperation,supervise,implement,agencies
eni	Enel,gazprom,statoil,petrobras,lokoil,rosneft ,ptt,exxonmobil,conocophillips,total	enel,gazprom,,statoil,petrobras,lukoil, ,rosneft,ptt,rwe, repsol,ypf
fold	crumbs,knead,ostrich,yolk,tortilla,yeast,zest,dough,carrots,substitutions	dough,spatula,folds,folded,whisk,lengthwise,folding,tablespoons,spoon,gently

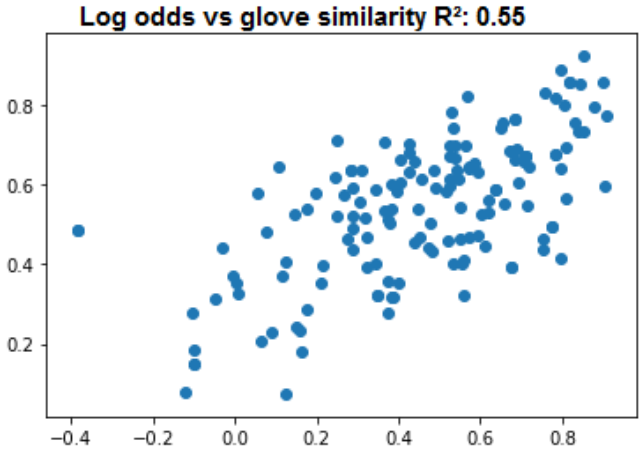
One has the sense that the two techniques do behave in different ways. The topical representation captures a more loosely linked semantic association. Glove similar words on the other hand seems to handle more accurately synonymy.

Furthermore, one can look in a more quantitative way at these results. For a random sample of 1000 words, how many closest words overlap? Below is the frequency histogram of overlap.



The mean is at 0.75. Therefore, we see that these two techniques output somehow different closest words, and seem to capture different information about the word at hand.

To better gain an intuition of how different the two techniques are, we plotted on the x-axis similarity as outputted by the topical technique, and on the y-axis the similarity score as outputted by glove. Although there is clear correlation between the two, there is still some topical information glove does not capture.



1.7.2 Reuters classification task

To confirm this intuition that indeed our topical representation achieves to capture more information than that simply in glove, we sought to understand if concatenating the 2 representation gives better results for our Reuters classification task.

Model	Micro F1	MacroF1	Recall micro	Precision micro
BOW	0,86	0,48	0,82	0,91
Glove 100	0,799	0,341	0,735	0,875
log odds descriptors 100 augmented	0,797	0,3714	0,746	0,856
Concat 100	0,824	0,395	0,779	0,874
Glove 200	0,826	0,399	0,779	0,879
log odds descriptors 200 augmented	0,825	0,399	0,776	0,881
Concat 200	0,838	0,441	0,803	0,877
Glove 300	0,837	0,429	0,793	0,886
log odds descriptors 300 augmented	0,839	0,442	0,805	0,876
Concat 300	0,852	0,443	0,803	0,909

There is a significant boost in performance as one concatenates the two sort of representation, arguing for the fact that the topical representation of words does capture information not included in glove.

1.8 Conclusion

We built a topical lexicon that maps each word to a 100, 200, 300-dimensional vector, representing the word's strength of association to 100, 200, and 300 selected topics. The

extended version of this topical lexicon – using glove features for missing topics or dimensions - with 100, 200, and 300 dimensions for roughly 50 000 words is present in the Data folder.

2. Applications of the topical lexicon for words: selecting core words of a language

2.1 Core words of a language

Core words of a language are defined as the words referring to pervading and universal concepts. Such words include ‘school’, ‘table’, ‘walk’ and the like. They do not convey any information other than these universal concepts, such as information about the style of the text, or the topic discussed. They are especially useful for language learners in that they constitute the very skeleton of a language, they are the building blocks for representing meaning. Therefore, identifying such core words on which to focus study has been a task with major stakes.

2.2 Extracting core words using topical representations

The usual way to identify these core words, is to take a corpus, not unlike our corpus of the New-York Times, compute raw frequencies for each word, and then select a subset of the x most common words than constitute say 80% of all words used in the corpus. The disadvantage of such a technique is that the corpus used usually is heavily biased towards a few specific topics - the New-York Times is biased towards World Politics for instance. This technique will yield not the core words of English, but more core words of English politics and international affairs. Also, core words of a language are not necessarily the most used, although they are very used. Words such as ‘leg’, or ‘mouth’, or other body parts, although essential, may have low frequencies in a large number of corpora. Any measure based solely on raw frequency misses the point.

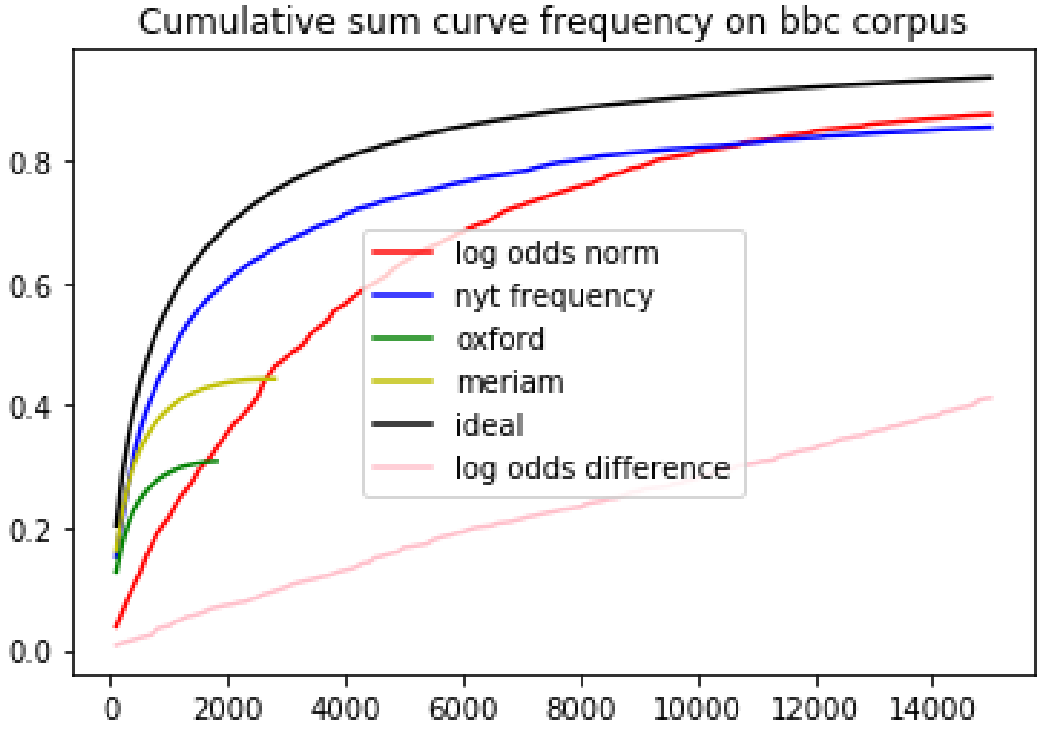
Here the approach we propose is different, and distances itself from frequency. It relies on the idea of topicality of words. True core words of English appear in roughly the same frequency no matter the corpus, and as such should not be heavily linked to any topic. To exploit this intuition, we will use our topical representation of words. So far, our technique projects each word in a d-dimensional space where each dimension is an association score to a specific topic. Building on this idea, core words of English should be vectors very close to the origin. Indeed, since they share the same distribution no matter the corpus, for each dimension they should have relatively low log odd scores. They are not linked to any specific topic.

Here we will use the extended version of our lexicon, since it allows to not deal with the problem of words that have more represented dimensions than others.

2.3 Evaluating Core words: cumulative frequency on out of sample corpus

A problem arises as we try to evaluate these models. How to say which list of core words is better than another? For this, we will rely again on the idea of frequency, although as we have stated before it is not ideal. The evaluation method will be to compute cumulative frequencies of the selected core words in a new corpus. How much of the corpus is captured by these lists of core words? This evaluation technique is tantamount to asking, if I know this vocabulary list, how comfortable will I be in this new setting? How many words will I be able to recognize and feel comfortable with in this new setting? As an out of sample corpus, we will use a large BBC dataset, with precomputed frequencies.

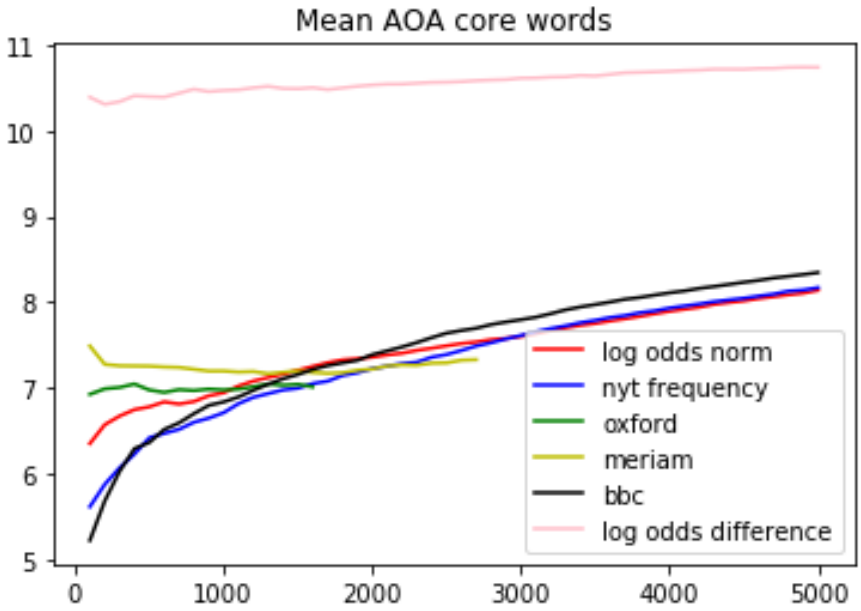
We selected 3 different baselines. The first is the core vocabulary as selected from the most frequent words in the New-York Times. The second and third are manually created lists of core words, one from the Oxford English Dictionary, the other from Meriam-Webster. We added a third technique based on our log odds score, computing the difference between the highest value in the vector and the lowest. However, this technique does not perform as well.



We can see that the log odds technique performs very well, better than the 3 baselines. It is very effective in capturing words with high frequencies, and does not overfit to a particular corpus. Arguably it is better than the technique based solely on frequency. Nevertheless, one can argue that this setting is not fair either for the Meriam Webster or the Oxford core words. Indeed, we selected our set of core words based on a news corpus, and we test it on another news corpus. The Meriam Webster and Oxford lexicons are aimed at general English, English used in all contexts: TV, books, daily life, and arguably news is a small subset of all English. There would be an need to evaluate this frequency on a wide array of corpora. We leave this area to further investigation by lack of time.

2.4 Evaluating Core words: mean Age of Acquisition

Age of Acquisition is an important variable in psycholinguistic, referring to the age at which a word is typically learned. Studies in psycholinguistics show that correlates well with some proxy measure of ease or relatedness, such as speed of reading words or of naming pictures. Furthermore, words that are more frequent, shorter, more familiar and refer to concrete concepts are learned earlier than more complex words. Therefore, it the age of acquisition measure can be used as a proxy for its core-ness. It is interesting to see which technique yields the lowest age of acquisition.



We can see here that the log odds technique normalizing the vector outperforms the 2 lexicons based on newspaper data, and is competitive with hand labeled core words lexicons.

3. Useful resources

In the course of our study, we were brought to build 2 interesting datasets. Although, because of lack of time, and because the study led us in different directions these 2 datasets were not exploited to their fullest potential, we would like to put them online nonetheless for research.

3.1 Meriam-Webster learner definitions

The Meriam-Webster learner page provides interesting tag for words, such as ‘technical’, or ‘medical’. And furthermore, the definition of a word as found in a dictionary can be used in several ways for NLP applications. Nevertheless, there is no freely available dictionary corpus, and therefore we decided to create our own.

To build this dictionary database, we scrawled the Meriam-Webster learner dictionary and obtained definitions for 36 000 words. The words and their definitions can be found in the Data folder.

3.2 PLOS Abstract / summary pair datasets

The PLOS – Public Library for Open Science – is a journal, mostly of medicine, that encourages its authors to publish in parallel of the scientific article, a lay summary of their work. This initiative is done to make science more accessible and to encourage pluridisciplinarity. Here we will use these summaries in a framework of text simplification or readability. By contrasting these lay summaries with the abstract of the article – the 2 are approximately of same length, and have the same purpose- it is possible to explore in an interesting way Readability.

Here we provide in the Data folder 20 000 odd abstract summary pairs, with the urls of the article.

4.0 Code and file description

4.1 Code

1 - meriam_crawler.py : Crawls the Meriam Webster learner dictionary and extracts definition

Parameters:
file_name : file_name with the words to look up, words, comma separated
max_time : max number of seconds before sending the next query, default 10, optional

Returns:
meriam_webster.csv: file with the word queried as the first column, and the definition as the second column

```
python meriam_crawler.py test_words.txt -max_time 5
```

2 - plos_crawler.py

Crawls the PLOS journal webstise dictionary and extracts abstract summary pairs

Parameters:
file_name : file_name with the words to look up, words, comma separated
max_time : max number of seconds befond sending the next query, default 10, optional

Returns:
Plos_database.csv: file with the url queried as the first column, the journal where the article was extracted as the second column, the abstract as the third, and the lay summary as the fourth

```
python plos_crawler.py test_plos.txt -max_time 5
```

3 – toolbox : Necessary functions to run the other codes, mainly include loading functions

4 – topical_lexicon: Computes the topical representation of words

Parameters:
dimensions : number of dimensions, from 0 to 300, default 100
threshold : threshold of counts. Words in the topical corpus chose count is below this threshold are not considered, default 10

Returns:
topical representation of words: dictionary whose keys are words, and values are the vector of its topical representation

5 – topical_lexicon_augmented : Computes the topical representation with an extension based on glove features

Parameters:
words_corpus: words whose representation is needed, if none is provided then computes

representation for all words present in glove, which is usually a lengthy process
dimensions: number of dimensions, from 0 to 300, default 100
threshold: threshold of counts. Words in the topical corpus chose count is below this threshold are not considered, default 10
alpha: value of the regularizer, default 1e-5
dimensions_glove: number of glove dimension representation with which to infer topical scores. To save space we only included the 100 dimensional distributed representation, default 100
return_r: Return the r_pearson and r_spearman coefficients for each topic

Return:
topical representation of words: dictionary whose keys are words, and values are the vector of its topical representation
if return_r = True: returns as well the r_pearson and r_spearman coefficients for each topic, as a list of tuple [r_pearson, r_spearman]

NOTE: need to download glove.6B.100d.txt and put it in the Data file to make the code word. Glove embeddings can be found at the following address:
https://nlp.stanford.edu/projects/glove/

4.2 Datasets

1 - Extended topical representation of 50 000 words

The 300-topic representation of words. To get partial topical representation of words, e.g. for only 100 dimensions, truncate the vector
/Data/300_dim_topical.csv

[Column 1] = word
[Column 2 to end] = score of association of word to the topic

2 - PLOS Abstract summary pairs

Lay summary and abstract pair from the PLOS journal
/Data/plos_abstract_summary.csv

[Column 1] = url
[Column 2] = field, i.e. the subjournal the article was extracted from
[Column 2] = abstract
[Column 3] = summary
Number of abstract / summary pairs : 20 000

3 - Meriam webster definition

Definitions of words from the Meriam Webster Learner dictionary
/Data/meriam_webster_definitions.csv

[Column 1] = word

[Column 2] = definition in Meriam Webster learner dictionary
Number of definitions: 36 000

4.3 External datasets

descriptors_count_dimensions: folder containing counts for each topic, i.e. descriptors

aoa: age of acquisition table

BBC_data: frequency count on bbc corpus

nyt_frequency: frequency count on New-York Times corpus

core_meriam: core words of English as stated by Meriam webster

core_oxford: core words of English as stated by Oxford Dictionary of English

descriptors_nyt: topic corresponding to each dimension