Natural Language Processing

Readability Estimation: A Recursive Formulation

Prasanna P & Vaishnavh N

November 28, 2014

Contents

Abstract

Readability of documents is generally estimated as a function of the features of the document such as average sentence length and average word syllable length. While documents have such readability estimates associated with them, words can also be associated with a quantity that approximates the ease of understanding them e.g., age of acquisition of words. Here we discuss a novel approach to estimate a word's difficulty. Our methodology does not require extensive external knowledge like accurate frequency estimates of the words and instead, bootstraps from co-occurrences of the word in the documents of a corpus.

1 Introduction

Why do we need word-difficulty estimates? The acquisition of words by a human is built upon his/her existing knowledge about other words. This results from the amount of inherent 'knowledge' contained in a word. Such contained knowledge in a word can be considered to be an estimate of the word's difficulty. The relevance of estimating it is two-fold. Firstly, there are many educational domains in which the vocabulary of growing children is enriched by exposure to banks of words at various ages. Knowing the difficulty of a word helps us fit the word into appropriate banks. Secondly, word-level estimates of readability can themselves be used in refining the estimates of readability of documents.For the sake of simplicity, we use the term *difficulty* of words in contrast to *readability* of documents, though both 'difficulty' and 'readability' essentially refer to similar ideas.

In this work, we present a method for estimating word difficulties using only a corpus as a knowledge base. Our method does not require external information about the number of syllables or the semantics of the words. We show on synthetic data that our algorithm works significantly better than a baseline approach that merely estimates word difficulties from the frequency of the word in the given corpus.

The circularity Examining the context words of the term `funds` in a corpus will yield us equally 'heavy' phrases like `share market`, `equity` and `investment`. Assuming we know how difficult such context words are, can we approximately pin-point the difficulty of the word `funds`? By thus knowing the difficulty of `funds`, In this case, we may note that the size of the word `funds` is misleading - it is as short as `star`, yet requires more insight to understand. Thus, contextual words may provide such missing information.

Let us further motivate the situation with the following examples. Consider the following extract from the scholarly journal Diacritics (1997) written by Judith Butler, professor of rhetoric and comparative literature at the University

of California at Berkeley [1]:

" The move from a structuralist account in which capital is understood to structure social relations in relatively homologous ways to a view of hegemony in which power relations ... "

The word `structuralist`, here, will almost never occur without knowledge-heavy words due to its conceptual richness. The following is a line from a Science news article:

" As per researchers, this works better by placing the nanostructures in a quasi-random pattern on the solar cells. "

Intuitively, we expect the difficulty estimate of the word `solar` to be skewed up, for it seems to occur with denser words like `quasi-random`. Consider the following weather report:

" Buffalo weather has since become an internet meme, since residents were amazed by how bad the weather actually got this time of year. "

The lack of words that require expertise to understand must indicate that words like `weather` are not as knowledge-rich as words like `structuralist` which are always accompanied by heavy contexts.

One could also recall the kinds of words used in poems and stories that are meant for children. One major inference from that above is that documents that are easy to read, use easier words. Thus, easier words tend to occur together. Therefore, when such texts are together considered as a corpus without information about the difficulty of the words themselves, we will be able to order these words across the spectrum of difficulty. For example,

Limitation? It is logical to suspect that the context might not always favour this: what if `weather` also occurs with words of meteorological depth? Our assumption is that we allow the values to be limited by the distribution of the documents in the corpus, for that is how well we can do. A corpus where `weather` occurs only in academic documents on meteorology will misguide the difficulty value of the word. However, for the given situation, this is acceptable. The burden lies on the corpus acquisition stage to retrieve a corpus that reflects average knowledge.

2 Literature Survey

Readability estimation discusses about how difficult a document is, for understanding. To extract this the difficulty level of words and other textual features are considered. There have been several publications in readability estimation

of documents in the last decade. Researchers have come up with readability estimation parameters like coh-metrics [6]. The public version of this gives 56 varied scores based on different textual indices. People have also experimented with several other parameters[2] like *word length, syllables, syntax etc.,* as a candidate for the formula.
The problem has been approached as a search for features[4] to be extracted from the documents, on which one can learn a model to predict the readability scores of the document. Usual experimentation is like identifying 'n' subjects and asking them to read through a document to rate its difficulty. The documents are represented as a vector of features. Later, the problem is solved as a regression over the features to estimate the difficulty level. [8] [9] [7] discusses these statistical methods. Having come up with different metrics for the estimation of readbility of documents people have also worked on something similar to voting or committee machines that combine more than one measure have also been looked at [3]. Also, few models [9] have considered the combination of language models and surface level features. We also looked at (Age of Acquisition) AoA [5] of a word that gives the expected age at which a person is likely to acquire the meaning of the word. This is directly proportional to the difficulty of the word.

2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative model. This assumes a topic distribution over the documents with the hyper parameter α and each document has a individual parameter θ which is a distribution over infinite topics. Further, each of the word has a distribution over the topics which is governed β. Learning the parameters θ, α and β is a problem of bayesian inference.

$$P(W,Z,\theta,\alpha,\beta) = \Pi\ P(W|Z,\theta,\beta,\alpha).P(Z|\theta\ \alpha).P(\theta|\alpha).P(\alpha).P(\beta)$$

Z is a vector of documents' topic. W is the words' topic.

3 Methodology

We first propose a simple generative model for the corpus, which describes our assumptions about how the difficult words in the corpus are distributed. This procedure is quite akin to Latent Dirichlet Allocation, as discussed earlier.We will assume that the difficulty of a word belongs to $\mathbb{R}$ for theoretical purposes. Practically, we will impose bounds.

For every document $d \in \mathcal{D}$:

- Sample n_d from Poisson(η). This defines the length of the document.

- Sample μ_d from $\mathcal{N}(\Delta)$ that is a normal distribution. Here Δ consists of two parameters, the mean and the standard deviation. This defines the expected difficulty of any word in the document.
- Sample σ_d from $\mathcal{N}(\Sigma)$ that is also a normal distribution.
- Sample σ_d from
- Sample n_d values from the distribution $\mathcal{N}(\Phi)$ where $\Phi = (\mu_d, \sigma_d)$. For each value, sample a word from the mapping defined by Θ.

The graphical model is shown below.

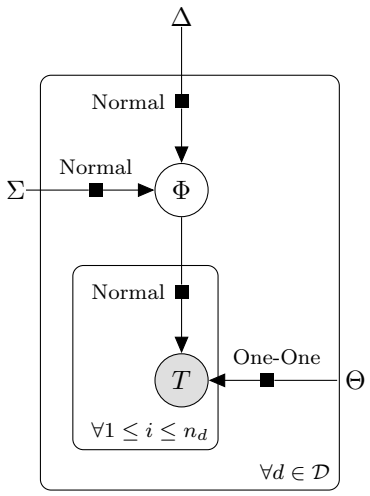

Figure 1: A directed factor graph for the normal generative model

Practical Issues and Assumptions First of all, all assumptions like exchangeability under the LDA model hold good here i.e., the joint distribution over the words are invariant to the ordering of the words.

In this model, further, we have assumed that every real value sampled from Φ corresponds to a single word in an infinite vocabulary. Practically during generation, we firstly curtail the ends of the normal distribution - any value sampled in the tail is mapped to the ends of the distribution. Furthermore, we discretize the support interval of this curtailed distribution in order to produce a corpus over a finite vocabulary with words repeating naturally.

We also assume that the readability of a document is an aggregate of the difficulty levels of all the words contained in it. The *readability* of any document in the corpus is expected to be around μ_Δ and the deviation from this is expected to be around σ_Δ. Further, within the documents, the difficulty of the words

can vary widely. This variation is dictated by an expectation of μ_Sigma with a deviation of σ_Σ. observed in the difficulty of the words seen in the document is expected to be Σ. When μ_Σ is high, the documents contain words that vary highly with respect to difficulty.

Alternative Model We also consider a model where the words in a document are sampled from an exponential distribution. We find this more intuitive because we rarely expect documents to contain only hard words. A text is usually 'supported' by many easy words and has interspersed harder words that contain the crux of the information. An exponential distribution, that theoretically assumes difficulty levels in the range $\mathbb{R}^+$ can model this behaviour. The appropriate figure can be found below.

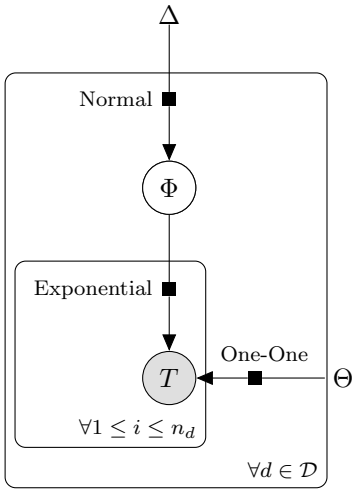

Figure 2: A directed factor graph for the exponential model

Here, Φ defines the parameter of the exponential distribution. Note that, we could use a **Gamma distribution** if we needed a greater degree of freedom.

3.1 Inference

Given the words T of all the documents, we are primarily interested in the parameters of Θ. These are respectively the difficulty 'level' that maps to a word. A harder word corresponds to a greater magnitude.

With T observed, we can define the likelihood as

$$\mathcal{L}(\Phi, \Theta; T) \quad = \quad \prod_{d \in \mathcal{D}} P(\mu_d|\Delta) P(\sigma_d|\Sigma) \prod_{w \in d} P(w|\mu_d, \sigma_d, \Theta)$$

Our aim is to find parameter settings that can maximize likelihood. To solve this maximization problem, we will find ML estimates for the parameters alternatingly.

3.1.1 Document Readability Estimation Step

In this step, the hyperparameter Θ is given i.e., we know how difficult the words are. We are required to estimate the difficulty levels of the document. Since Θ is known, we can replace w with the real number that it corresponds to, θ_w. Then, the maximization for a single document boils to maximizing:

$$\begin{aligned} P(T, \Phi|\Theta, \Delta) &\propto \prod_{d \in \mathcal{D}} P(\sigma_d|\Sigma) P(\mu_d|\Delta) \prod_{w \in d} P(w|\mu_d, \sigma, \Theta) \\ &\propto P(\mu_d|\Delta) P(\sigma_d|\Sigma) \prod_{w \in d} P(\theta_w|\mu_d, \sigma) \end{aligned}$$

Estimating μ_d : We first fix σ_d while estimating μ_d. Now, notice that $P(\mu_d|\Delta)$ is a normal prior and $P(\theta_w|\mu_d, \sigma_d)$ is a normal distribution with known variance. Since these are **conjugate**, we can state that the maximum likelihood estimate of μ_d is:

$$\frac{\frac{\mu_\Delta}{\sigma_\Delta^2} + \frac{\sum_{w \in d} \theta_w}{\sigma^2}}{\frac{1}{\sigma_\Delta^2} + \frac{n_d}{\sigma^2}}$$

Estimating σ_d : We do not discuss this here because an assumption we will further make about σ_Δ will void the need for this.

3.1.2 Word Difficulty Estimation step

Here we assume that the parameters Φ are known. We have t rediscover the values of Θ. Considering a specific word w,

$$\begin{aligned} P(T, \Theta|\Phi, \Delta) &\propto \prod_{d \in \mathcal{D}_w} P(w|\mu_d, \sigma_d, \Theta) \\ &\propto \prod_{d \in \mathcal{D}_w} P(\theta_w|\mu_d, \sigma_d) \end{aligned}$$

Here $\mathcal{D}_w$ refers to the multi-set of documents corresponding to every occurence of w. This is a simple MLE estimate, where in we need to find θ_w, the difficulty corresponding to w. The product of the probabilities will be an exponential with the power equal to $\sum_{\mathcal{D}_w} (\theta_w - \mu_d)^2$. The value of θ_w that maximizes this is:

$$\frac{\sum_{\mathcal{D}_w} \mu_d}{|\mathcal{D}_w|}$$

Practical Issues An important question we have to address is the assignment of values to Δ and Σ. Should we *infer* them? Should we assume them as constants? It is possible to do inference. However, the process gets complicated with having to estimate many parameters. Instead we assume that the value is given. A further assumption is that the corpus difficulty distribution is uniform i.e, $\sigma_\Delta = \infty$. Therefore, we will be estimating μ_d as:

$$\frac{\sum_{w \in d} \theta_w}{n_d}$$

The EM interpretation The above iterative estimation algorithm can be seen as an expectation-maximization process. We can compare the difficulty levels of documents to the cluster assignments of points in k-means. The difficulty levels of words are the cluster centroids. In the expectation step, we estimate the difficulty of the documents as an average of the difficulty levels of words. In the maximization step, we estimate the difficulty of the words as an average of the difficulty of the documents in which it occurs.

The eigen-value interpretation Determining the Θ is akin to determining the eigenvector of a matrix. We will define $\boldsymbol{D}$ as a document-term matrix with as many rows as there are documents and as many columns as there are words in the vocabulary. Now, define $\tilde{\boldsymbol{D}}$ as the row-stochastic document-term matrix where each row is proportional to the corresponding row in D. Next, let $\boldsymbol{G}$ be a row-stochastic term-document matrix that is equivalent to $\tilde{\boldsymbol{D^T}}$. Let $\boldsymbol{\phi}_i$ and $\boldsymbol{\theta}_i$ be column vectors corresponding to the document readabilities and word difficulties in the ith iteration of our algorithm.

The document readability estimation step can be rewritten as:

$$\boldsymbol{\mu}_i = \tilde{\boldsymbol{D}} \boldsymbol{\phi}_{i-1}$$

The word difficulty estimation step can be rewritten as:

$$\boldsymbol{\phi}_i = \tilde{\boldsymbol{G}} \boldsymbol{\mu}_i$$

The above equations imply that $\boldsymbol{\phi}_i = \tilde{\boldsymbol{G}} \tilde{\boldsymbol{D}} \boldsymbol{\phi}_{i-1}$. That is, on convergence:

$$\boldsymbol{\phi} = \tilde{\boldsymbol{G}} \tilde{\boldsymbol{D}} \boldsymbol{\phi}$$

In other words, $\boldsymbol{\phi}$ is an eigenvector of $\tilde{\boldsymbol{G}} \tilde{\boldsymbol{D}}$!

3.2 Exponential distribution

For the exponential distribution, we do not have a neat prior that returns another exponential distribution. Thus, we limit ourselves to assuming again that Δ is uniform distribution across all values. In such a case, estimating the mean document difficulty given by β_d boils down to maximizing:

$$P(T, \Phi|\Theta, \Delta) \propto \prod_{w \in d} P(\theta_w|\beta_d)$$

The ML estimate is equal to :

$$\frac{\sum_{w \in d} \theta_w}{n_d}$$

To estimate word difficulties, we will be maximizing:

$$\begin{aligned} P(T, \Phi|\Theta, \Delta) &\propto \prod_{d \in \mathcal{D}_w} P(\theta_w|\beta_d) \\ &\propto \prod_{d \in \mathcal{D}_w} \exp(-\frac{\theta_w}{\beta_d}) \end{aligned}$$

However, minimizing that would be equivalent to setting all θ_w to 0. We would like to avoid this trivial solution. Instead we will aim to maximize the whole equation:

$$\begin{aligned} P(T, \Phi|\Theta, \Delta) &\propto \prod_{w \in T} \exp\left(-\theta_w \sum_{d \in \mathcal{D}_w} \frac{1}{\beta_d}\right) \\ &\propto \exp\left(-\sum_{w \in T} \theta_w \sum_{d \in \mathcal{D}_w} \frac{1}{\beta_d}\right) \end{aligned}$$

Now, we can add an additional constraint that $\sum_{w \in T} \theta_w^2$ is constant which will ensure a non-zero assignment to all the word-difficulties. The above optimization can now be seen as maximizing the dot-product between $\vec{\theta}$ and another vector defined by the $\sum_{d \in \mathcal{D}_w} \frac{1}{\beta_d}$ sums. We can thus specify $\vec{\theta}$ to be a unit-vector along the second vector.

Note If we do not want to use a continuous distribution such as Φ in generating the words, a parameterzied geometric distribution that can be considered is the hyper-geometric distribution.

4 Experiments and Results

We run experiments on synthetically generated corpora. More specifically, we assume there are as many buckets as there words in the assumed vocabulary. The real-value returned by the normal distribution is mapped to one of these buckets with the tail ends mapping to the end buckets. We could have alternatively mapped the tail ends to one of the buckets uniformly. Another practical issue we have to consider here is the unboundedness of the difficulties as they are being operated on by the matrices. This could cause computational issues with underflows/overflows. We overcome this by rescaling the difficulty values

to fit between 0 and 1 during the algorithm. However, **is this still equivalent to converging to the dominant eigenvector?**

We perform inference on the corpus and also compare it against the eigenvectors for the normal distribution model.

While generating documents, we assume that difficulty levels for words can take integer values from 0 uptil $|V|-1$ where V is the vocabulary. All parameters are presented at this scale. The mean of the poisson distribution is set at 40.

Evaluation We determine the correlation ($\hat{\rho}$) between the inferred difficulty values for the words and the actual difficulty values that were assumed while generating. ρ refers to the correlation of the eigenvector that is the best. The n_e denotes the position of the eigenvector as ordered by the eigenvalues, e. Also, we report the **magnitude** of the correlation because it might converge to -1 or 1 depending on the direction of the eigenvector.

Baseline The baseline measure (ρ_b) that we use is the frequency count of the words in the synthetic corpus i.e., more frequent words are easier. **However,** this is not a true baseline when we do not assume an exponential distribution.

Effect of increasing σ_Δ (document-level variance)

$\lvert\mathcal{D}\rvert$	$\lvert V\rvert$	μ_Δ	σ_Δ	ρ_b	$\hat{\rho}$	ρ	n_e	e
10000	1000	500	10	0.013	-	0.013	2	(1)
10000	1000	500	50	0.005	0.974	0.974	2	(1,0.08)
10000	1000	500	100	0.005	0.993	0.993	2	(1,0.22)
10000	1000	500	300	0.002	0.998	0.998	2	(1, 0.67)

- $\rho \gg \rho_b$ is surprising. We would expect ρ_b to be much better. We cross-verified this by computing a similar baseline on **Rueters** corpus in NLTK and assuming that **AoA** values are the actual estimates of the difficulty of the words. Computing baseline frequency correlation with AoA yields 0.0147 which seems to follow the general trend of a near-zero correlation.
- The fact that the performance becomes better with more σ_Δ can be explained by the fact that the corpus becomes more uniform in distribution. That is, the corpus has both as many easy documents as there as difficult documents with a greater σ_Δ and this helps the inference. Our algorithm has been suited to infer from such situations better.
- **Why does the algorithm converge to the second dominant eigenvalue?** We suspected that this could probably be because of the *rescaling* that we do after every step. However, even if that operation weren't there the algorithm converges to the second dominant eigenvector.

Effect of increasing document count

100	1000	500	200	0.0031	0.0089	0.79	2	(1, 0.67)
1000	1000	500	200	0.021	0.983	0.983	2	(1, 0.51)
2000	1000	500	200	0.0029	0.991	0.991	2	(1, 0.5)
10000	1000	500	200	0.003	0.998	0.998	2	(1, 0.51)

As expected an increase in the corpus size helps the inference. However, for a small corpus size the matrices still have information about the difficulty level which doesn't manifest in the inference though.

Effect of increasing vocabulary size

$\|\mathcal{D}\|$	$\|V\|$	μ_Δ	σ_Δ	ρ_b	$\hat{\rho}$	ρ	n_e	e
10000	400	200	80	0.000092	0.977	0.977	2	(1, 0.14)
10000	1000	500	200	0.0016	0.998	0.998	2	(1, 0.51)
10000	2000	1000	400	0.241	0.5852	0.76	1	(1, 0.67)
10000	10000	5000	2000	0.38	0.658	0.658	1	(1, 0.85)
10000	20000	10000	4000	0.300	0.48	0.48	2	(1, 0.89)

A number of interesting phenomena occur here.

- Firstly, the performance of our model declines. This is expected as our model would do better when there are more samples. This follows from the central limit theorem.
- The baseline performance increases! This is probably because when the vocabulary is small and a relatively large number of samples are made all the words appear with seemingly random frequency not correlated with the difficulty. This will happen due to documents of various difficulties being sampled. When the number of samples is relatively small, the *pattern* that encourages easier words to occur more

On the whole, we see encouraging results on the synthetic data generated by our model.[1]

4.1 Experiments on Real Dataset

Our initial experimentations involved a merge of the Rueters, Gutenberg and Brown corpora available on NLTK. Due to computational difficulties in providing an exhaustive comparision of results, we present only results for Rueters here. In Rueters, we consider each `paragraph` as a standalone document and we removed the **stopwords** [2].

[1] We are currently working on the results of the exponential model. Despite an error in our coding, we seem to be getting really good results. We are investigating this and hence avoiding presenting them here.

[2] Perhaps, not removing it would have given better values.

As discussed previously we use AoA values normalized between 0 and 1 as the correct difficulty estimates. Our first observations is that our models do not perform well. Some of our initial heuristics gave, at the best, an increase of **0.01** in the correlation. But our initial heuristics (which involved using momentum variables) used to proceed in the positive direction for the first few iterations and the progressively start worsening. The lack of convergence was a concern which eventually drove us to producing a rigorous mathematical model.

The average number of words in a document is around 10.5 with standard deviation 38. There are 11887 documents and 28714 words. The average corpus difficulty when scaled to the vocabulary size comes to 8619 with a standard deviation of 1076. From a sample of 100 such documents we estimated $\mu_\Sigma = 36179$. Running the algorithm on this dataset doesn't converge at all! Simulation with these parameters on a synthetic dataset generated by our model gives a poor converged correlation of 0.364 but with a baseline of 0.25. This bloated baseline implies that our model does not represent the dataset very well, and hence cannot be compared. (The baseline of Rueters is less than 0.02!)

However, using occurences of just 10 words from Rueters which are the most frequent in the corpus yields a correlation of 0.3658: but this doesn't compare against the 0.72 baseline that corresponds to these words. [3]

An analysis for the first 10 eigenvectors (of the matrices corresponding to the normal distribution model) for Rueters corpus shows the **4**th principal eigenvector to have a correlation of 0.22 which is the highest. Interestingly, in some ways it is very close to the second principal eigenvector - $1, 0.677, 0.65, 0.611, 0.48$ are the eigenvalues in order.

5 Conclusions

We summarize our discussion until here.

- We have proposed two generative model and two inference techniques to handle the same. However, we make strong assumptions in the models - such as a uniform distribution in the difficulty of the documents. This may not be true, as for example, **Rueters** has **a normal distribution** over the document difficulties computed from AoA.

[3] It is likely that the exponential model will work better here.

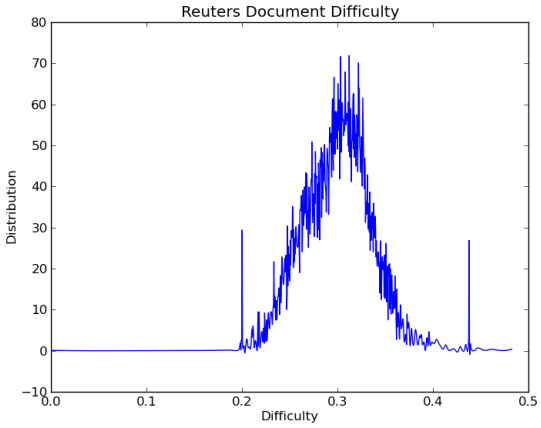

Figure 3: Document Readability Distribution

- We have proposed a two kinds of distributions: a normal distribution and an exponential. We expected the exponential distribution to be valid as most documents usually have many stop words which are "easy". However, removing the stop words in the Rueters dataset and plotting these distributions often showed a single mode around which the words' difficulty levels were centered.
- We have provided an eigenvector representation of the solution for the approximate form of our algorithm
- Our inference algorithms work very well on synthetic data despite them being generated from a non-uniform prior ($\sigma_\Delta < \infty$), which is against our assumption.
- One of the eigenvectors of the complete Rueters dataset has a correlation of 0.22 with AoA. This is promising. The poor performance of the convergence mechanism can be attributed to the presence of many eigenvectors with similar eigenvalues. Moreover, the dataset has a very **low variance** with respect to document difficulty. This could explain the poor performance too.

6 Conclusions and Future Work

The results indicate that our inference technique is rigorous enough. The concern however is that the hypothetical model may not reflect real datasets. Our effort must be in engineering this model for practical relevance. The following

are some of the ideas we wish to explore further based on the results we have seen.

- We have assumed a generic model over the distribution of difficulty over words. It would be interesting to see how this changes when we assume different topics. Each topic must induce a diffirent distribution over the words, coupled with the difficulty.
- We have not used a hypergeometric distribution to model the discrete distribution over the words. It might be a more rigorous approach.
- We could consider acquiring a dataset which presents a uniform distribution over the difficulty of documents, as our proposed methodology works on that assumption.
- We could also consider updating the hyperparameter for the distribution of the corpus difficulty instead assuming a fixed value.
- It would be useful to understand how the power iteration method converges to vectors other than the second dominant eigenvalue.

7 Some Ideas on Readability Estimation

We report below other ideas that we included in our initial proposal for the project.

7.1 Inverse Readablility Estimation

Is it possible to determine the age or at least, the *category* to which a user belongs, with the knowledge of the kind of texts that the person reads or writes comfortably, or based on the readability score that the person gives to various texts? This might be helpful in recommendation systems, where one could suggest, say, blog articles to the user based on the kind of language that is used in the blogs that they usually read.

7.2 Incorporating User Expertise

Readability is often blind to the user's familiarity with words. It gives a generalized idea of how difficult the document might be for a second language learner to read. What if we want to estimate how difficult an article on `biology` seems to a computer engineer? And an article on `networking` is to a doctor? Can we somehow model this user knowledge and bias the results of the readability accordingly?

References

[1] The Bad Writing Contest. http://denisdutton.com/bad_writing.htm.

[2] Scott A Crossley, David B Allen, and Danielle S McNamara. Text readability and intuitive simplification: A comparison of readability formulas. *Reading in a foreign language*, 23(1):84–101, 2011.

[3] Scott A Crossley, David F Dufty, Philip M McCarthy, and Danielle S McNamara. Toward a new readability: A mixed model approach. In *Proceedings of the 29th annual conference of the Cognitive Science Society*, pages 197–202, 2007.

[4] Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad. A comparison of features for automatic readability assessment. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 276–284. Association for Computational Linguistics, 2010.

[5] Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. Age-of-acquisition ratings for 30,000 english words. *Behavior Research Methods*, 44(4):978–990, 2012.

[6] Danielle S McNamara, Max M Louwerse, and Arthur C Graesser. Coh-metrix: Automated cohesion and coherence scores to predict text readability and facilitate comprehension. *Unpublished Grant proposal, University of Memphis, Memphis, Tennessee*, 2002.

[7] Sarah E Petersen and Mari Ostendorf. A machine learning approach to reading level assessment. *Computer speech & language*, 23(1):89–106, 2009.

[8] Emily Pitler and Ani Nenkova. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195. Association for Computational Linguistics, 2008.

[9] Luo Si and Jamie Callan. A statistical model for scientific readability. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 574–576. ACM, 2001.