A Dictionary of Wisdom and Wit: Learning to Extract Quotable Phrases*

Michael Bendersky
Dept. of Computer Science
University of Massachusetts
Amherst, MA
bemike@cs.umass.edu

David A. Smith
Dept. of Computer Science
University of Massachusetts
Amherst, MA
dasmith@cs.umass.edu

Abstract

Readers suffering from information overload have often turned to collections of pithy and famous quotations. While research on large-scale analysis of text reuse has found effective methods for detecting widely disseminated and famous quotations, this paper explores the complementary problem of detecting, from internal evidence alone, which phrases are *quotable*. These quotable phrases are memorable and succinct statements that people are likely to find useful outside of their original context. We evaluate quotable phrase extraction using a large digital library and demonstrate that an integration of lexical and shallow syntactic features results in a reliable extraction process. A study using a *reddit* community of quote enthusiasts as well as a simple corpus analysis further demonstrate the practical applications of our work.

1 Introduction

Readers have been anxious about information overload for a long time: not only since the rise of the web, but with the earlier explosion of printed books, and even in manuscript culture (Blair, 2010). One traditional response to the problem has been excerpting passages that might be useful outside their original sources, copying them into personal commonplace books, and publishing them in dictionaries such as *Bartlett's Familiar Quotations* or the *Oxford Dictionary of Quotations*. Even on the web, collection of quotable phrases continues to thrive[1], as evidenced by the popularity of quotation websites such as *BrainyQuote* and *Wikiquote*.

According to a recent estimate, there are close to 130 million unique book records in world libraries today (Taycher, 2010). Many of these books are being digitized and stored by commercial providers (e.g., Google Books and Amazon), as well as non-profit organizations (e.g., Internet Archive and Project Gutenberg).

As a result of this digitization, the development of new methods for preserving, accessing and analyzing the contents of literary corpora becomes an important research venue with many practical applications (Michel et al., 2011). One particularly interesting line of work in these large digital libraries has focused on detecting *text reuse*, i.e., passages from one source that are quoted in another (Kolak and Schilit, 2008).

In contrast, in this paper we explore the modeling of phrases that *are likely to be* quoted. This phrase modeling is done based on internal evidence alone, regardless of whether or not the phrase actually *is* quoted in existing texts.

We call such potential quotation a **quotable phrase** – a meaningful, memorable, and succinct statement that can be quoted without its original context. This kind of phrases includes aphorisms, epigrams, maxims, proverbs, and epigraphs.

* "The book is a dictionary of wisdom and wit..." *(Samuel Smiles, "A Publisher and His Friends")* This and all the subsequent quotations in this paper were discovered by the proposed quotable phrase extraction process.

[1] "Nothing is so pleasant as to display your worldly wisdom in epigram and dissertation, but it is a trifle tedious to hear another person display theirs." *(Kate Sanborn, "The Wit of Women")*

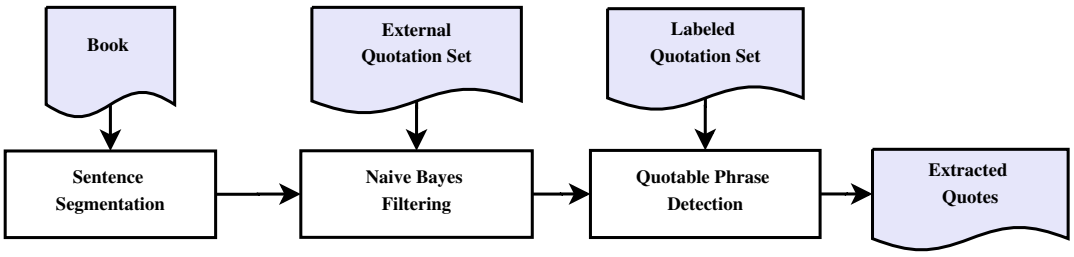

Figure 1: Diagram of the quotable phrase extraction process.

A computational approach to quotable phrase extraction has several practical applications. For instance, it can be used to recommend new additions to existing quotable phrase collections, especially focusing on lesser read and studied authors and literary works[2]. It can also generate quotable phrases that will serve as catchy and entertaining previews for book promotion and advertisement[3].

In this work, we describe such a computational approach to quotable phrase extraction. Our approach leverages the Project Gutenberg digital library and an online collection of quotations to build a *quotable language model*. This language model is further refined by a supervised learning algorithm that combines lexical and shallow syntactic features.

In addition, we demonstrate that a computational approach can help to address some intriguing questions about the nature of quotability. What are the lexical and the syntactic features that govern the quotability of a phrase? Which authors and books are highly quotable? How much variance is there in the perceived quotability of a given phrase?

The remainder of this paper is organized as follows. In Section 2 we provide a detailed description of the entire process of quotable phrase extraction. In Section 3 we review the related work. In Sections 4 and 5 we evaluate the quotable phrase extraction process, and provide some corpus quotability analysis. We conclude the paper in Section 6.

2 Quotable Phrase Extraction

There are three unique challenges that need to be addressed in the design of the process of quotable phrase extraction. The first challenge stems from the fact that the boundaries of potential quotes are often ambiguous. A quotable phrase can consist of a sentence fragment, a complete sentence, or a passage of text that spans several sentences.

The second challenge is that the occurrence of quotable phrases is a rare phenomena in literary corpora. A randomly selected book passage is unlikely to be quotable without any additional context.

The third challenge is related to the syntax and semantics of quotable phrases. For instance, consider the phrase

> "Evil men make evil use of the law, though the law is good, while good men die well, although death is an evil." *(Thomas Aquinas, "Summa Theologica")*

and contrast it with

> "Of the laws that he can see, the great sequences of life to death, of evil to sorrow, of goodness to happiness, he tells in burning words." *(Henry Fielding, "The Soul of a People")*

While both of these phrases share a common vocabulary (*law*, *death*, *good* and *evil*), the latter sentence contains unresolved pronouns (*he*, twice) that make it less amenable to quotation out of context.

Accordingly, we design a three-stage quotable phrase extraction process, with each stage corresponding to one of challenges described above. The diagram in Figure 1 provides a high-level overview of the entire extraction process on a single book. Next, we provide a brief description of this diagram. Then, in the following sections, we focus on individual stages of the extraction process.

To address the first challenge of quote boundary detection, at the first stage of the extraction process

[2] "There is life in a poet so long as he is quoted..." *(Sir Alfred Comyn Lyall, "Studies in Literature and History")*

[3] As an example, see the "Popular Highlights" feature for Kindle e-books in the *Amazon* bookstore.

(*Sentence Segmentation*) we segment the text of the input book into sentences using an implementation of the Punkt sentence boundary detection algorithm (Kiss and Strunk, 2006). In an initial experiment, we found that 78% of the approximately 4,000 quotations collected from the *QuotationsPage*[4] consist of a single sentence. From now on, therefore, we make a simplifying assumption that an extracted quotable phrase is confined within the sentence boundaries.

The second processing stage (*Naïve Bayes Filtering*) aims to address the second challenge (the rarity of quotable phrases) and significantly increases the ratio of quotable phrases that are considered as candidates in the final processing stage (*Quotable Phrase Detection*). To this end, we use a set of quotations collected from an external resource to build a *quotable language model*. Only sentences that have a sufficiently high likelihood of being drawn from this language model are considered at the final processing stage.

For this final processing stage (*Quotable Phrase Detection*), we develop a supervised algorithm that focuses on the third challenge, and analyzes the syntactic structure of the input sentences. This supervised algorithm makes use of structural and syntactic features that may effect phrase quotability, in addition to the vocabulary of the phrase.

2.1 Naïve Bayes Filtering

In order to account for the rarity of quotable phrases in the book corpus, we use a filtering approach based on a pre-built *quotable language model*. Using this filtering approach, we significantly reduce the number of sentences that need to be considered in the supervised quotable phrase detection stage (described in Section 2.2). In addition, this approach increases the ratio of quotable phrases considered at the supervised stage, addressing the problem of the sparsity of positive examples.

To build the quotable language model, we bootstrap the existing quotation collections on the web. In particular, we collect approximately 4,000 quotes on more than 200 subjects from the *QuotationsPage*. This collection provides a diverse set of high-quality quotations on subjects ranging from *Laziness* and *Genius* to *Technology* and *Taxes*.

[4] `www.quotationspage.com`

Then, we build two separate unigram language models. The first one is the quotable language model, which is built using the collected quotations ($\mathcal{L}_Q$). The second one is the background language model, which is built using the entire book corpus ($\mathcal{L}_C$). Using these language models we compute a log-likelihood ratio for each processed sentence $\mathbf{s}$, as

$$LLR_{\mathbf{s}} = \sum_{w \in \mathbf{s}} \ln \frac{p(w|\mathcal{L}_Q)}{p(w|\mathcal{L}_C)}, \quad (1)$$

where the probabilities $p(w|\cdot)$ are computed using a maximum likelihood estimate with add-one smoothing.

A sentence $\mathbf{s}$ is allowed to pass the filtering stage if and only if $LLR_{\mathbf{s}} \in [\alpha, \beta]$, where α, β are positive constants[5]. The lower bound on the $LLR_{\mathbf{s}}$, α, requires the sentence to be highly *probable* given the quotable language model $\mathcal{L}_Q$. The upper bound on the $LLR_{\mathbf{s}}$, β, filters out sentences that are highly *improbable* given the background language model $\mathcal{L}_C$.

Finally, the sentences for which $LLR_{\mathbf{s}} \in [\alpha, \beta]$ are allowed to pass through the Naïve Bayes filter. They are forwarded to the next stage, in which a supervised quotable phrase detection is performed.

2.2 Supervised Quotable Phrase Detection

In a large corpus, a supervised quotable phrase detection method needs to handle a significant number of input instances (in our corpus, an average-sized book contains approximately 2,000 sentences). Therefore, we make use of a simple and efficient perceptron algorithm, which is implemented following the description by Bishop (2006).

We note, however, that the proposed supervised detection method can be also implemented using a variety of other binary prediction techniques. In an initial experiment, we found that more complex methods (e.g., decision trees) were comparable to or worse than the simple perceptron algorithm.

Formally, we define a binary function $f(\mathbf{s})$ which determines whether an input sentence $\mathbf{s}$ is a quotable (q) or a non-quotable ($\overline{q}$) phrase, based on:

$$f(\mathbf{s}) = \begin{cases} q & \text{if } \mathbf{w}\mathbf{x}_{\mathbf{s}} > 0 \\ \overline{q} & \text{else,} \end{cases} \quad (2)$$

[5] In this work, we set $\alpha = 1, \beta = 25$. This setting is done prior to seeing any labeled data.

Feature	Description
	Lexical
LLR	Sentence log-likelihood ratio (Eq. 1)
#word	Number of words in $\mathbf{s}$.
#char	Number of characters in $\mathbf{s}$.
wordLenAgg	Feature for each aggregate Agg of word length in $\mathbf{s}$.
	Agg = {*min, max, mean*}
#capital	Number of capitalized words in $\mathbf{s}$.
#quantifier	Number of universal quantifiers in $\mathbf{s}$ (from a list of 13 quantifiers, e.g., *all*, *whole*, *nobody*).
#stops	Number of common stopwords in $\mathbf{s}$.
beginStop	True if $\mathbf{s}$ begins with a stopword, False otherwise.
hasDialog	True if $\mathbf{s}$ contains at least one of the three common dialog terms {*say*, *says*, *said*}.
#abstract	Number of abstract concepts (e.g., *adventure*, *charity*, *stupidity*) in $\mathbf{s}$.
	Punctuation
hasP	Five features to indicate whether punctuation of type P is present in $\mathbf{s}$.
	P = {*quotations, parentheses, colon, dash, semi-colon*}.
	Parts of Speech
#POS	Four features for the number of occurrences of part-of-speech POS in $\mathbf{s}$.
	POS = {*noun, verb, adjective, adverb, pronoun*}.
hasComp	True if $\mathbf{s}$ contains a comparative adjective or adverb, False otherwise.
hasSuper	True if $\mathbf{s}$ contains a superlative adjective or adverb, False otherwise.
hasPP	True if $\mathbf{s}$ contains a verb in past participle, False otherwise.
#IGSeq[i]	Count of the POS sequence with the i-th highest $IG(X, Y)$ (Eq. 3) in $\mathbf{s}$.

Table 1: Description of the quotability features that are computed for each sentence $\mathbf{s}$.

where $\mathbf{x_s}$ is a vector of *quotability features* computed for the sentence $\mathbf{s}$, and $\mathbf{w}$ is a weight vector associated with these features. The weight vector $\mathbf{w}$ is updated using stochastic gradient descent on the perceptron error function (Bishop, 2006).

Since Eq. 2 demonstrates that the supervised quotable phrase detection can be formulated as a standard binary classification problem, its success will be largely determined by an appropriate choice of feature vector $\mathbf{x_s}$. As we are unaware of any previous work on supervised detection of quotable phrases, we develop an initial set of easy-to-compute features that considers the lexical and shallow syntactic structure of the analyzed sentence.

2.3 Quotability Features

A decision about phrase quotability is often subjective; it is strongly influenced by personal taste and circumstances[6]. Therefore, the set of features that we describe in this section is merely a coarse-grained approximation of the true intrinsic qualities of a quotable phrase. Nevertheless, it is important to note that these features do prove to be beneficial in the context of the quote detection task, as is demonstrated by our empirical evaluation in Section 5.

Table 1 details the quotability features, which are divided into 3 groups: *lexical*, *punctuation-based* and *POS-based* features. All of these features are conceptually simple and can be efficiently computed even for a large number of input sentences.

Some of these features are inspired by existing text analysis tasks. For instance, work on readability detection for the web (Kanungo and Orr, 2009; Si and Callan, 2001) examined features which are similar to the *lexical* features in Table 1. *Parts of speech* features (e.g., the presence of comparative and superlative adjectives and adverbs) have been extensively used for sentiment analysis and opinion mining (Pang and Lee, 2008).

In addition, we use a number of features based on simple hand-crafted word lists. These lists include word categories that could be potential indicators of quotable phrases such as universal quantifiers (e.g., *all*, *everyone*) and abstract concepts[7].

[6] "One man's beauty another's ugliness; one man's wisdom another's folly." *(Ralph Waldo Emerson, "Essays")*

[7] For abstract concept modeling we use a list of 176 abstract nouns available at www.englishbanana.com.

The novel features in Table 1 that are specifically designed for quotable phrase detection are based on part of speech sequences that are highly indicative of quotable (or, conversely, non-quotable) phrase (features `#IGSeq[i]`). In order to compute these features we first manually label a validation set of 500 sentences that passed the Naïve Bayes Filtering (Section 2.1). Then, we apply a POS tagger to these sentences, and for each POS tag sequence of length n, seq_n, we compute its information gain

$$IG(X, Y) = H(X) - H(X|Y). \quad (3)$$

In Eq. 3, X is a binary variable indicating the presence or the absence of seq_n in the sentence, and $Y \in \{q, \overline{q}\}$.

We select k sequences seq_n with the highest value of $IG(X, Y)$[8]. We use the count in the sentence of the sequence seq_n with the i-th highest information gain as the feature `#IGSeq[i]`. Intuitively, the features `#IGSeq[i]` measure how many POS tag sequences that are indicative of a quotable phrase (or, conversely, indicative of a non-quotable phrase) the sentence contains.

3 Related Work

The increasing availability of large-scale digital libraries resulted in a recent surge of interest in computational approaches to literary analysis. To name just a few examples, Genzel et al. (2010) examined machine translation of poetry; Elson et al. (2010) extracted conversational networks from Victorian novels; and Faruqui and Padó (2011) predicted formal and informal address in English literature.

In addition, computational methods are increasingly used for identification of complex aspects of writing such as humor (Mihalcea and Pulman, 2007), double-entendre (Kiddon and Brun, 2011) and sarcasm (Tsur et al., 2010). However, while successful, most of this work is still limited to an analysis of a single aspect of writing style.

In this work, we propose a more general computational approach that attempts to extract quotable phrases. A quotability of a phrase can be affected by various aspects of writing including (but not limited to) humor and irony[9], use of metaphors[10], and hyperbole[11].

It is important to note that our approach is conceptually different from the previous work on paraphrase and quote detection in book corpora (Kolak and Schilit, 2008), news stories (Liang et al., 2010) and movie scripts (Danescu-Niculescu-Mizil et al., 2012). While this previous work focuses on mining popular and oft-used quotations, we are mainly interested in discovering quotable phrases that might have never been quoted by others.

Number of books	$21,492$
Number of authors	$8,889$
Total sentences	$4.45 \cdot 10^7$
After Naïve Bayes filtering	$1.75 \cdot 10^7$

Table 2: Summary of the Project Gutenberg corpus.

4 Experimental Setup

To evaluate the quotable phrase extraction process in its entirety (see Figure 1), we use a collection of Project Gutenberg (*PG*) books[12] – a popular digital library containing full texts of public domain books in a variety of formats. In particular, we harvest the entire corpus of 21,492 English books in textual format from the *PG* website.

The breakdown of the *PG* corpus is shown in Table 2. The number of detected sentences in the *PG* corpus exceeds 44 million. Roughly a third of these sentences are able to pass through the Naïve Bayes Filtering (described in Section 2.1) to the supervised quotable phrase detection stage (Section 2.2).

For each of these sentences, we compute a set of lexical and syntactic features described in Section 2.3. For computing the features that require the part of speech tags, we use the MontyLingua package (Liu, 2004).

[8] In this work, we set $n = 3, k = 50$. This setting is done prior to seeing any labeled data.

[9] "To be born with a riotous imagination and then hardly ever to let it riot is to be a born newspaper man." *(Zona Gale, "Romance Island")*

[10] "If variety is the spice of life, his life in the north has been one long diet of paprika." *(Fullerton Waldo, "Grenfell: Knight-Errant of the North")*

[11] "The idea of solitude is so repugnant to human nature, that even death would be preferable." *(William O.S. Gilly, "Narratives of Shipwrecks of the Royal Navy; between 1793 and 1849")*

[12] `http://www.gutenberg.org/`

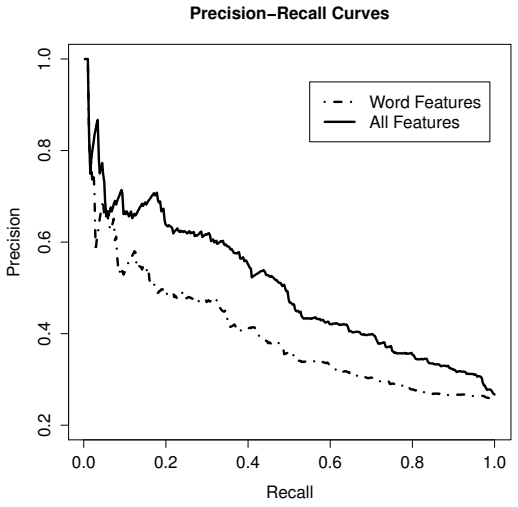

Figure 2: Prec. vs. recall for quotable phrase detection.

We find that the extraction process shown in Figure 1 is efficient and scalable. On average, the entire process requires less than ten seconds per book on a single machine.

The complete set of extracted quotable phrases and annotations is available upon request from the authors. In addition, the readers are invited to visit `www.noisypearls.com`, where a quotable phrase from the set is published daily.

5 Evaluation and Analysis

5.1 Naïve Bayes Filtering Evaluation

In the Naïve Bayes Filtering stage (see Section 2.1) we evaluate two criteria. First, we measure its ability to reduce the number of sentences that pass to the supervised quotable phrase detection stage. As Table 2 shows, the Naïve Bayes Filtering reduces the number of these sentences by more than 60%.

Second, we evaluate the recall of the Naïve Bayes Filtering. We are primarily interested in its ability to reliably detect quotable phrases and pass them through to the next stage, while still reducing the total number of sentences.

For recall evaluation, we collect a set of $2,817$ previously unseen quotable phrases from the *Goodreads* website[13], and run them through the Naïve Bayes Filtering stage. $2,262$ (80%) of the quotable phrases pass the filter, indicating a high quotable phrase recall.

[13] `http://www.goodreads.com/quotes`

1	`#abstract`	$+91.64$
2	`#quantifier`	$+61.67$
3	`hasPP`	-60.34
4	`#IGSeq[16]` *(VB IN PRP)*	$+39.71$
5	`#IGSeq[6]` *(PRP MD VB)*	-38.78
6	`#adjective`	$+37.71$
7	`#IGSeq[14]` *(DT NN VBD)*	-36.88
8	`#verb`	$+35.22$
9	`beginStop`	$+31.73$
10	`#noun`	$+29.63$

Table 3: Top quotability features.

Based on these findings, we conclude that the proposed Naïve Bayes Filtering is able to reliably detect quotable phrases, while filtering out a large number of non-quotable ones. It can be further calibrated to reduce the number of non-quotable sentences or to increase the quotable phrase recall, by changing the setting of the parameters α and β, described in Section 2.1. In the remainder of this section, we use its output to analyze the performance of the supervised quotable phrase detection stage.

5.2 Quotable Phrase Detection Evaluation

To evaluate the performance of the supervised quotable phrase detection stage (see Section 2.2) we randomly sample 1,500 sentences that passed the Naïve Bayes Filtering (this sample is disjoint from the sample of 500 sentences used for computing the `IGTagSeq` feature in Section 2.3). We annotate these sentences with q (*Quotable*) and $\overline{q}$ (*Non-Quotable*) labels.

Of these sentences, 381 (25%) are labeled as *Quotable*. This ratio of quotable phrases is much higher than what is expected from a non-filtered content of a book, which provides an indication that the Naïve Bayes Filtering provides a relatively balanced input to the supervised detection stage.

We use this random sample of 1,500 labeled sentences to train a perceptron algorithm (as described in Section 2.2) using 10-fold cross-validation. We train two variants of the perceptron. The first variant is trained using only the lexical features in Table 1, while the second variant uses all the features.

Figure 2 compares the precision-recall curves of these two variants. It demonstrates that using the

Popular	$\Uparrow \geq 10$	12
Upvoted	$1 \leq \Uparrow \leq 10$	34
No upvotes	$\Uparrow \leq 0$	14
	$\mathbf{p}(\Uparrow > \mathbf{0}) =$	**.77**

Table 4: Distribution of *reddit* upvote scores.

syntactic features based on punctuation and part of speech tags significantly improves the precision of quote phrase detection at all recall levels. For instance at the 0.4 recall level, it can improve precision by almost 25%.

Figure 2 also shows that the proposed method is reliable for high-precision quotable phrase detection. This is especially important for applications where recall is given less consideration, such as book preview using quotable phrases. The proposed method reaches a precision of 0.7 at the 0.1 recall level.

It is also interesting to examine the importance of different features for the quotable phrase detection. Table 3 shows the ten highest-weighted features, as learned by the perceptron algorithm on the entire set of 1,500 labeled examples.

The part of speech features `#IGTagSeq[i]` occupy three of the positions in the Table 3. It is interesting to note that two of them have a high *negative weight*. In other words, some of the POS sequences that have the highest information gain (see Eq. 3) are sequences that are indicative of non-quotable phrases, rather than quotable phrases.

The two highest-weighted features are based on handcrafted word lists (`#abstract` and `#quantifier`, respectively). This demonstrates the importance of task-specific features such as these for quotability detection.

Finally, the presence of different parts of speech in the phrase (nouns, verbs and adjectives), as well as their verb tenses, are important features. For instance, the presence of a verb in past participle (`hasPP`) is a strong *negative* indicator of phrase quotability.

5.3 The *reddit* Study

As mentioned in Section 2.3, the degree of the phrase quotability is often subjective, and therefore its estimation may vary among individuals. To validate that our quotability detection method is not biased by our training data, and that the detected quotes will have a universal appeal, we set up a verification study that leverages an online community of quote enthusiasts.

For our study, we use *reddit*, a social content website where the registered users submit content, in the form of either a link or a text post. Other registered users then upvote or downvote the submission, which is used to rank the post.

Specifically, we use the *Quotes subreddit*[14], an active *reddit* community devoted to discovering and sharing quotable phrases. At the time of this writing, the *Quotes* subreddit has more than 12,000 subscribers. A typical post to this subreddit contains a single quotable phrase with attribution. Any *reddit* user can then upvote or downvote the quote based on its perceived merit.

To validate the quality of the quotes which were used for training the perceptron algorithm, we submitted 60 quotes, which were labeled as quotable by one of the authors, to the *Quotes* subreddit. At most one quote per day was submitted, to avoid negative feedback from the community for "spamming".

Table 4 presents the upvote scores of the submitted quotes. An upvote score, denoted $\Uparrow$, is computed as

$$\Uparrow = \#\text{ upvotes } - \#\text{ downvotes.}$$

Table 4 validates that the majority of the quotes labeled as quotable, were also endorsed by the *reddit* community, and received a non-negative upvote score. As an illustration, in Table 5, we present five quotes with the highest upvote scores. Anecdotally, at the time of this writing, only one of the quotes in Table 5 (a quote by Mark Twain) appeared in web search results in contexts other than the original book.

5.4 Project Gutenberg Corpus Analysis

In this section, we briefly highlight an interesting example of how the proposed computational approach to quotable phrase extraction can be used for a literary analysis of the *PG* digital library. To this end, we train the supervised quotable phrase detection method using the entire set of 1,500 manually labeled sentences. We then run this model over all the 17.5 million sentences that passed the Naïve Bayes

[14] `http://www.reddit.com/r/quotes`

Quote	⇑
"One hour of deep agony teaches man more love and wisdom than a whole long life of happiness." *(Walter Elliott, "Life of Father Hecker")*	**49**
"As long as I am on this little planet I expect to love a lot of people and I hope they will love me in return." *(Kate Langley, Bosher, "Kitty Canary")*	**43**
"None of us could live with an habitual truth-teller; but thank goodness none of us has to." *(Mark Twain, "On the Decay of the Art of Lying")*	**40**
"A caged bird simply beats its wings and dies, but a human being does not die of loneliness, even when he prays for death." *(George Moore, "The Lake")*	**33**
"Many will fight as long as there is hope, but few will go down to certain death." *(G. A. Henty, "For the Temple")*	**30**

Table 5: Five quotes with the highest upvote scores on *reddit*.

(a) Authors			**(b) Books**		
1	Henry Drummond	.045	*1*	"Friendship" (Hugh Black)	.113
2	Ella Wheeler Wilcox	.041	*2*	"The Dhammapada" (Translated by F. Max Muller)	.112
3	S. D. Gordon	.040	*3*	"The Philosophy of Despair" (David Starr Jordan)	.106
4	Andrew Murray	.038	*4*	"Unity of Good" (Mary Baker Eddy)	.097
5	Ralph Waldo Emerson	.037	*5*	"Laments" (Jan Kochanowski)	.084
6	Orison Swett Marden	.034	*6*	"Joy and Power" (Henry van Dyke)	.079
7	Mary Baker Eddy	.031	*7*	"Polyeucte" (Pierre Corneille)	.078
8	'Abdu'l-Bahá	.029	*8*	"The Forgotten Threshold" (Arthur Middleton)	.078
9	John Hartley	.029	*9*	"The Silence" (David V. Bush)	.077
10	Rabindranath Tagore	.028	*10*	"Levels of Living" (Henry Frederick Cope)	.075

Table 6: Project Gutenberg (a) authors and (b) books with the highest quotability index.

filtering stage, and retain only the sentences that get positive perceptron scores (Eq. 2).

This procedure yields 701,418 sentences, which we call *quotable phrases* in the remainder of this section. These quotable phrases are less than 2% of the entire Project Gutenberg corpus; however, they still constitute a sizable collection with some potentially interesting properties.

We propose a simple example of a literary analysis that can be done using this set of quotable phrases. We detect books and authors that have a high *quotability index*, which is formally defined as

$$QI(x) = \frac{\text{\# quotable phrases}(x)}{\text{\# total sentences}(x)},$$

where x is either a book or an author. To ensure the statistical validity of our analysis, we limit our attention to books with at least 25 quotable phrases and authors with at least 5 books in the *PG* collection.

Using this definition, we can easily compile a list of authors and books with the highest quotability index (see Table 6). An interesting observation is that many of the authors and books in Table 6 deal with religious themes: Christianity (e.g., Mary Baker Eddy, S. D. Gordon), Bahá'ísm ('Abdu'l-Bahá) and Buddhism ("The Dhammapada"). This is not surprising considering the figurative language common in the religious prose[15].

6 Conclusions

As the number of digitized books increases, a computational analysis of literary corpora becomes an active research field with many practical applications. In this paper, we focus on one such application: extraction of quotable phrases from books. Quotable phrase extraction can be used, among other things, for finding new original quotations for dictionaries and online quotation repositories, as well as for generating catchy previews for book advertisement.

We develop a quotable phrase extraction process that includes sentence segmentation, unsupervised sentence filtering based on a *quotable language model*, and a supervised quotable phrase detection using lexical and syntactic features. Our evaluation demonstrates that this process can be used for high-precision quotable phrase extraction, especially in applications that can tolerate lower recall. A study using a *reddit* community of quote enthusiasts as well as a simple corpus analysis further demonstrate

[15] "If a man speaks or acts with an evil thought, pain follows him, as the wheel follows the foot of the ox that draws the carriage."(*"The Dhammapada", translated by F. Max Muller*)

the practical applications of our work.

7 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant IIS-0910884 and in part by ARRA NSF IIS-9014442. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Ann M. Blair. 2010. *Too Much to Know: Managing Scholarly Information before the Modern Age*. Yale University Press.

Cristian Danescu-Niculescu-Mizil, Justin Cheng, Jon Kleinberg, and Lillian Lee. 2012. You had me at hello: How phrasing affects memorability. In *Proc. of ACL*, page To appear.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. In *Proc. of ACL*, pages 138–147.

Manaal Faruqui and Sebastian Padó. 2011. "I thou thee, thou traitor": predicting formal vs. informal address in English literature. In *Proceedings of ACL-HLT*, pages 467–472.

Dmitriy Genzel, Jakob Uszkoreit, and Franz Och. 2010. "Poetic" Statistical Machine Translation: Rhyme and Meter. In *Proc. of EMNLP*, pages 158–166.

Tapas Kanungo and David Orr. 2009. Predicting the readability of short web summaries. In *Proc. of WSDM*, pages 202–211.

Chloe Kiddon and Yuriy Brun. 2011. That's What She Said: Double Entendre Identification. In *Proc. of ACL-HLT*, pages 89–94.

T. Kiss and J. Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Okan Kolak and Bill N. Schilit. 2008. Generating links by mining quotations. In *Proc. of 19th ACM conference on Hypertext and Hypermedia*, pages 117–126.

Jisheng Liang, Navdeep Dhillon, and Krzysztof Koperski. 2010. A large-scale system for annotating and querying quotations in news feeds. In *Proc. of SemSearch*.

Hugo Liu. 2004. Montylingua: An end-to-end natural language processor with common sense. Available at: `web.media.mit.edu/~hugo/montylingua`.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The

Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Rada Mihalcea and Stephen Pulman. 2007. Characterizing humour: An exploration of features in humorous texts. In *Proc. of CICLing*, pages 337–347.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Luo Si and Jamie Callan. 2001. A statistical model for scientific readability. In *Proc. of CIKM*, pages 574–576.

Leonid Taycher. 2010. Books of the world, stand up and be counted! All 129,864,880 of you. *Inside Google Books blog*.

Oren Tsur, Dimitry Davidov, and Avi Rappoport. 2010. ICWSM–A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proc. of ICWSM*, pages 162–169.